

3100000011734

TUGAS AKHIR

(OE 1701)

**PENGEMBANGAN PERANGKAT LUNAK PENUNJANG
PRAKTIKUM MATA KULIAH ANALISA KEANDALAN
DAN RESIKO (OE 1520)**



RSK
005.1
Für
p-1
1998

Oleh :

PERPUSTAKAAN I T S	
Tgl. Terima	18-4-99
Terima Dari	H
No. Agenda Prg.	8767

EDI DJOKO FIRJANTO

NRP : 4390.100.008

**JURUSAN TEKNIK KELAUTAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1998**

**PENGEMBANGAN PERANGKAT LUNAK PENUNJANG
PRAKTIKUM MATA KULIAH ANALISA KEANDALAN
DAN RESIKO (OE 1520)**

TUGAS AKHIR

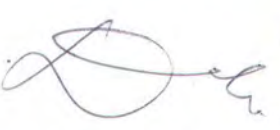
**Diajukan Guna Memenuhi Salah Satu Syarat
Untuk Menyelesaikan Studi Program Sarjana
Pada
Jurusan Teknik Kelautan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember
Surabaya**

Surabaya, 7 Agustus 1998

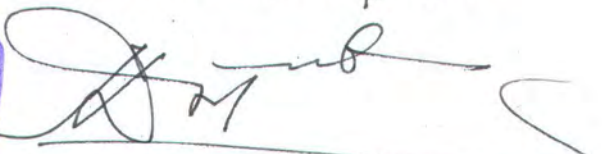
Mengetahui/Menyetujui :

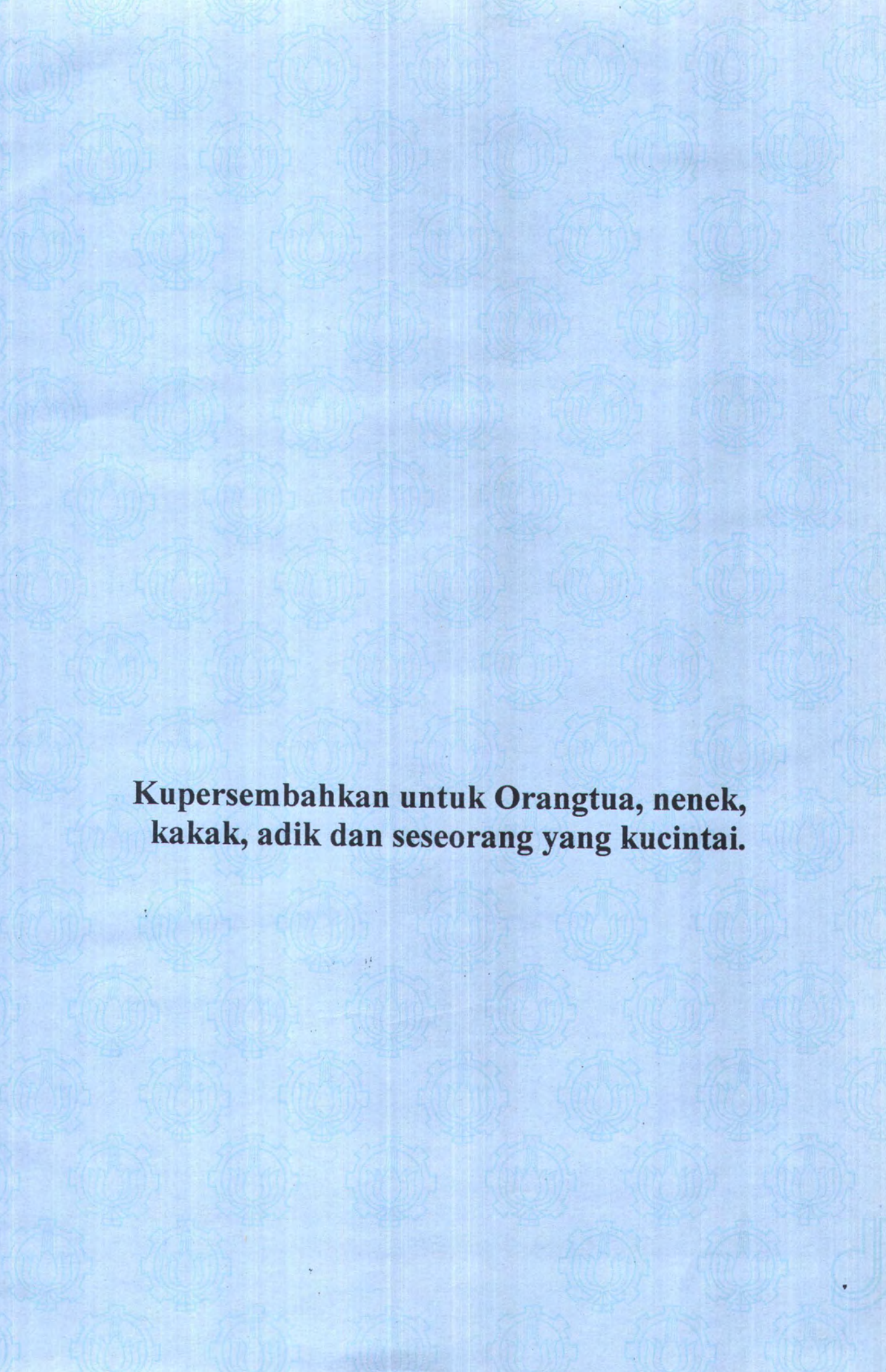
Dosen Pembimbing I

Dosen Pembimbing II


Ir. Daniel M. Rosyid PhD
NIP. : 131 782 038




Ir. Eko Budi Djatmiko, MSc. PhD
NIP. : 131 407 592



**Kupersembahkan untuk Orangtua, nenek,
kakak, adik dan seseorang yang kucintai.**



ABSTRAK

Dan darimana saja kamu berangkat maka palingkanlah wajahmu kearah Masjidil Haram, dan dimana saja kamu berada maka palingkanlah wajahmu kearahnya, agar tidak ada hujjah bagi manusia atas kamu, kecuali oran-orang yang zalim diantara mereka, maka janganlah kamu takut kepada mereka dan takutlah kepada-Ku saja, dan agar Kusempurnakan nikmat-Ku atasmu, dan agar kamu mendapat petunjuk. (QS Al Baqarah:150)

Abstrak

Tugas akhir ini membahas tentang pengembangan perangkat lunak untuk mendukung praktikum mata kuliah Analisa Keandalan Dan Resiko (OE. 1520) di Teknik Kelautan, yang akan memberikan panduan kepada mahasiswa dalam melakukan praktikum yang tersusun secara sistematis, mulai dari persiapan sampai selesai. Penulisan dititikberatkan pada Computer Aided Learning sebagai metode belajar-mengajar dengan menggunakan alat bantu komputer. Bahasa pemrograman yang dipakai untuk menulis perangkat lunaknya adalah Borland Delphi 3, karena lebih terstruktur dan dapat memberikan tampilan yang menarik. Untuk setiap praktikum disediakan studi kasus yang bersesuaian dengan metode pembahasan. Praktikum pertama menganalisa keandalan balok akibat momet maksimum dengan menggunakan metode AFOSM. Praktikum kedua menganalisa sambungan Y akibat in-plane bending moment dan out-plane bending moment dengan menggunakan Simulasi Monte Carlo. Praktikum terakhir mencari peluang kegagalan top event dari sistem listrik dan menentukan komponen-komponen penyebab kegagalan top event yang diselesaikan dengan Fault Tree Analysis.





KATA PENGANTAR

KATA PENGANTAR

Bismillahir Rohannir Rohiim.

Puji syukur Alhamdulillah kepada Yang Mahatunggal, Allah SWT, atas segala rahmat, hidayat dan kekuatan yang telah diberikan kepada penulis. Hanya atas izin dan kehendak-Nyalah Tugas Akhir ini dapat terselesaikan meskipun masih jauh dari kesempurnaan.

Tugas Akhir berjudul Pengembangan Perangkat Lunak Penunjang Praktikum Analisa Keandalan dan Resiko penulis selesaikan sebagai salah satu syarat untuk menyelesaikan studi di Jurusan Teknik Kelautan, FTK – ITS. Selain itu diharapkan Tugas Akhir ini mampu memberikan manfaat bagi pengembangan Praktikum Mata Kuliah Analisa Keandalan dan Resiko.

Akhirnya penulis menyadari sebagai manusia biasa yang tak lepas dari kesalahan, mengharapkan kritik dan saran yang membangun demi kebaikan bersama. Semoga tugas akhir ini dapat memberikan manfaat yang sebesar-besarnya bagi pembaca dan dapat membantu mahasiswa untuk melakukan praktikum MK. Analisa Keandalan dan Resiko serta mencapai tujuan yang diharapkan.

Surabaya, 17 Agustus 1998

Penulis



UCAPAN TERIMA-KASIH

UCAPAN TERIMA-KASIH

Penulis ingin mengucapkan segala rasa terima kasih yang sedalam-dalamnya dan penghargaan yang setulus-tulusnya kepada berbagai pihak yang telah memberikan dukungan, dorongan, bantuan dan bimbingan atas selesainya pengerjaan Tugas Akhir ini terutama kepada:

1. Bapak, Ibu, Nenek dan saudaraku yang sangat tulus cinta kasihnya kepada penulis, serta doa-doanya yang selalu mengalir.
2. Bapak Ir Daniel Muhammad Rosyid Ph.D dan Ir. Eko Budi Djatmiko MSc. Ph.D selaku dosen pembimbing yang dengan segala kesabaran, ketulusan dan keikhlasannya membimbing penulis selama penyelesaian tugas akhir ini.
3. Bapak Ir Daniel Muhammad Rosyid Ph.D selaku dosen wali yang selalu dengan sabar dan kasih dalam membimbing penulis selama masa kuliah.
4. Bapak Ir. Eko Budi Djatmiko MSc. Ph.D selaku Ketua Jurusan Teknik Kelautan beserta staf pengajar dan staf administrasi Teknik Kelautan.
5. Saudara Rofiq, Dasril Meidy, Wasis Sol Saleh, Jo Hakim Ambon, Hamid Elektro atas ilmunya.
6. Teman-teman Kelautan '90 Agus Petra, Ali Abah Sex, Cak Sastro Bachtiar, Budi Gimbun, Umar, Cholis, Doni Pendekar, Wagey Omo, Hadi, Pak De Heri, Mbah Miftah, Hendratmoko, Mbah Isnan, Cak Zaenuri, Mbak Nunung, Putut Rochani, Pak Dosen Rudi, Abang Tigor, Pak Dosen Suntuyo, Santo Rejes, Koh Tarno, Si Gendut Widyo atas kebersamaan, kesetiakawanan dan keakrabannya.
7. Penghuni apartemen Asempayung 29 Marutho, Q-Mic, Bendhol, Bung Imron, Singo, Agus Dhono, Hendro, Simbah Joko, Koh Wa'an, Bambang Ome, Piping Jemblunk, Edi Kempud, dll atas rame-ramenya.
8. Konco-konco kontrakan Bagus, Widodo, Budhi dan Iwan.
9. Penghuni Lantai 4 Kangmas Kepis, Iwan Kawat, Budi Onthonk, Gering Kurus Kerontang, Kanchonet, Cak Anas, Cak Arinta, Kasbulatov, Poouji, Adi dll atas semua senang dan susah yang telah dihadapi bersama.
10. Kelautan 91 Dik Dosen Roni, Gatot dan Asih, Veralin, Ice, Baron, Dullah, Bondet, Handy, Ayik, Day Suday Muday, Heri Artis, dll atas dukungannya.
11. Adikku Veralinix yang sangat menghargai dan sangat mempercayai penulis, terima-kasih atas semua smurfnya.
12. Siapa saja yang tidak dapat penulis sebutkan di sini tanpa mengurangi rasa hormat.
13. Mbak Arum.

Atas segala kebaikan dan bantuannya penulis panjatkan doa, semoga Allah berkenan membalas segala kebaikan yang penulis terima dengan kebaikan yang berlipat ganda dan semoga bantuannya mendapat nilai di sisiNya.



DAFTAR ISI

DAFTAR ISI

Lembar Pengesahan	i
Abstrak	ii
Kata Pengantar	iii
Ucapan Terima-kasih	iv
Daftar Isi	v

BAB I PENDAHULUAN

I.1. Latar Belakang	I - 1
I.2. Tujuan Penulisan	I - 2
I.3. Batasan permasalahan	I - 2
I.4. Metodologi Penulisan	I - 3

BAB II DASAR TEORI

II.1. Umum	II - 1
II.2. Teori Pendidikan	II - 2
II.3. Teori Analisa Keandalan dan Resiko	
II.3.1. Konsep Probabilitas	II - 7
II.3.2. Konsep Keandalan	
II.3.2.1. Konsep Ketidakpastian	II - 12
II.3.2.2. Perubah Acak dan Fungsi Kerapatan Peluang	II - 13
II.3.2.3. Diagram Interferensi	II - 16
II.3.2.4. Distribusi Peluang	II - 19
II.3.2.5. Metode Mean Value First Order Second Moment	II - 21
II.3.2.6. Indeks Keandalan Hashofer dan Lind	II - 25
II.3.2.7. Simulasi Monte Carlo	II - 27
II.3.2.8. Fault Tree Analysis	II - 32

BAB III PENGEMBANGAN METODE PRAKTIKUM

III.1. Penentuan Jenis Praktikum	III - 1
III.2. Penentuan studi kasus	III - 2
III.2.1.Kasus I	III - 4
III.2.2.Kasus II	III - 5
III.2.3.Kasus III	III - 7
III.3. Penentuan Prosedur Praktikum	III - 8
III.3.1.Perumusan Masalah	III - 9
III.3.2.Pengumpulan Informasi	III - 10
III.3.3.Penentuan Hipotesa	III - 11
III.3.4.Pelaksanaan Praktikum	III - 11
III.3.5.Penyusunan Laporan	III - 12
III.4. Penentuan Penyelesaian Studi Kasus	III - 12
III.4.1.Penyelesaian Studi Kasus I	III - 13
III.4.2.Penyelesaian Studi Kasus II	III - 15
III.4.3.Penyelesaian Studi Kasus III	III -17

BAB IV PRAKTIKUM

IV.1. Praktikum I	IV - 2
IV.1.1. Tujuan Praktikum I	IV - 2
IV.1.2. Petunjuk Pelaksanaan	IV - 3
IV.2. Praktikum II	IV - 5
IV.2.1. Tujuan Praktikum II	IV - 5
IV.2.2. Petunjuk Pelaksanaan	IV - 5
IV.3. Praktikum III	IV - 8
IV.3.1. Tujuan Praktikum III	IV - 8
IV.3.2. Petunjuk Pelaksanaan	IV - 8
IV.4. Penulisan Laporan	IV - 10

BAB V KESIMPULAN DAN SARAN

V - 1

DAFTAR PUSTAKA

LAMPIRAN A

LAMPIRAN B



BAB I PENDAHULUAN

Bacalah atas nama Tuhanmu yang menciptakan (segala makhluk). Yang menciptakan manusia dari segumpal darah. Bacalah; dan Tuhanmu adalah yang paling Mahamulia. Yang mengajarkan dengan kalam. Ia (Allah) yang mengajar manusia apa yang mereka tidak tahu. (QS. Al Alaq 1-5)

BAB I

PENDAHULUAN

I.1. LATAR BELAKANG

Mata Kuliah Analisa Keandalan dan Resiko yang dulunya mata kuliah pilihan (TL 1607) kemudian dirubah menjadi mata kuliah wajib (OE 1520), adalah merupakan salah satu mata kuliah yang sangat penting di Teknik Kelautan. Mata Kuliah ini berhubungan dan mendukung beberapa mata kuliah lain dalam proses perencanaan sistem-sistem rekayasa.

Mahasiswa sering merasakan kesulitan dalam memahami materi mata kuliah ini bila hanya disajikan secara numerik tanpa melakukan aplikasi-aplikasi, latihan-latihan atau praktikum dengan menggunakan perangkat lunak.

Dari alasan-alasan tersebut dibutuhkan sistem pengajaran dan sarana penunjang yang lebih memadai agar tercapai proses belajar mengajar yang lebih berkualitas.

I.2. TUJUAN PENULISAN

Tujuan penulisan Tugas Akhir ini adalah untuk membantu dan lebih memotivasi mahasiswa dalam mempelajari mata kuliah Analisa Keandalan dan Resiko dengan menyediakan satu set perangkat lunak yang dapat digunakan oleh mahasiswa untuk melakukan praktikum.

Diharapkan mahasiswa lebih memahami substansi-substansi mata kuliah Analisa Keandalan dan Resiko dan dapat menyelesaikan persoalan-persoalan rekayasa umum dibantu dengan perangkat lunak.

Dengan praktikum yang dilakukan diharapkan mahasiswa dapat lebih aktif baik secara individu maupun kelompok dan dapat merumuskan, menganalisa, menyelesaikan masalah-masalah dari studi kasus yang dipilih.

I.3. BATASAN PERMASALAHAN

Penulisan tugas akhir ini lebih ditekankan pada pengembangan perangkat lunak penunjang praktikum, oleh sebab itu permasalahan dibatasi pada:

- a. Penentuan jenis dan tujuan praktikum.
- b. Menyusun langkah-langkah/prosedur praktikum.
- c. Penyelesaian studi kasus yang dipilih.



Metode yang digunakan untuk menyelesaikan studi kasus, yaitu:

- a. Advanced First Order Second Moment (AFOSM).
- b. Simulasi Monte Carlo.
- c. Fault Tree Analysis (FTA).

I.4. METODOLOGI PENULISAN

BAB I. PENDAHULUAN

Berisi tentang latar belakang penulisan, tujuan penulisan, batasan-batasan permasalahan yang ditetapkan dalam penulisan dan metodologi penulisan yang dipakai dalam menyusun Tugas Akhir ini.

BAB II. DASAR TEORI

Mengemukakan beberapa dasar teori yang dipakai sebagai pijakan untuk penyusunan Tugas Akhir ini, yaitu Teori umum, Teori Pendidikan dan Praktikum serta Teori Analisa Keandalan dan Resiko.

BAB III. PENGEMBANGAN METODE PRAKTIKUM

Membahas tentang pengembangan praktikum mulai dari penentuan jenis praktikum, tujuan praktikum, studi kasus, prosedur praktikum dan penyelesaian studi kasusnya.

BAB IV. PRAKTIKUM

Menjelaskan tentang praktikum-praktikum yang akan dilakukan mulai Praktikum I, Praktikum II dan Praktikum III.

BAB V. PENUTUP

Berisi tentang kesimpulan dan saran yang dapat ditarik dari penulisan Tugas Akhir ini.

LAMPIRAN

Lampiran yang diberikan mengenai software praktikum dan petunjuk pelaksanaan praktikum.



BAB II DASAR TEORI

Dialah yang menciptakan matahari bersinar cemerlang dan bulan bercahaya terang, dan ditetapkan-Nya tempat-tempat peredarannya agar kamu dapat mengetahui bilangan tahun dan dapat menghitung (waktu). Allah tidak menciptakan itu semua kecuali dengan kebenaran; dijelaskan-Nya keterangan-keterangan untuk kaum yang (mau) mengetahui. (QS. Yunus:5)

BAB II

DASAR TEORI

II.1. UMUM

Pendidikan merupakan salah satu kegiatan manusia untuk lebih meningkatkan sumber daya dan kualitasnya, dengan pendidikan ini manusia akan dapat lebih mempermudah hidupnya dan lebih survival dalam mengatasi gejala-gejala alam yang diluar kemampuannya. Dalam rangka pencapaian tujuan-tujuan pendidikan perlu ditentukan metode pengajaran yang paling cocok untuk lebih menarik minat dan hasrat dari mahasiswa dalam mempelajarinya.

Praktikum merupakan salah satu metode pengajaran yang melatih mahasiswa untuk menerapkan ilmu yang didapatnya dalam masalah-masalah yang telah ditentukan. Dengan praktikum mahasiswa akan belajar merumuskan masalah, menentukan metode penyelesaian sekaligus menyelesaikannya secara tepat dan efisien. Selain itu di akhir praktikum mahasiswa diharuskan menulis laporan praktikum untuk mengetahui sejauh mana mahasiswa menguasai teori-teori yang mendasari praktikum tersebut dan sejauh mana mahasiswa dapat menerapkan teori-teori dasar dalam menyelesaikan masalah.

II.2. TEORI PENDIDIKAN

Salah satu program untuk mengembangkan metode belajar-mengajar ilmu-ilmu rekayasa adalah dengan menekankan pada keterlibatan mahasiswa dalam proses belajar yang aktif melalui kegiatan-kegiatan yang berorientasikan pada *discovery* dan/atau *inquiry*. Dasar filsafat ini ialah mahasiswa akan dimotivasi lebih baik bila ia terlibat secara langsung dalam proses belajar melalui kegiatan-kegiatan *discovery*.

Kegiatan *discovery* ialah suatu kegiatan atau pelajaran yang dirancang sedemikian rupa sehingga mahasiswa dapat menemukan konsep-konsep dan prinsip-prinsip melalui proses mentalnya sendiri. Kegiatan *inquiry* adalah perluasan proses-proses *discovery* yang digunakan dalam cara yang lebih dewasa [Amien (1988)].

Menurut Jerome Bruner, seorang profesor psikologi dari Harvard University [Amien (1988)] menyatakan beberapa keuntungan metode *discovery* sebagai berikut:

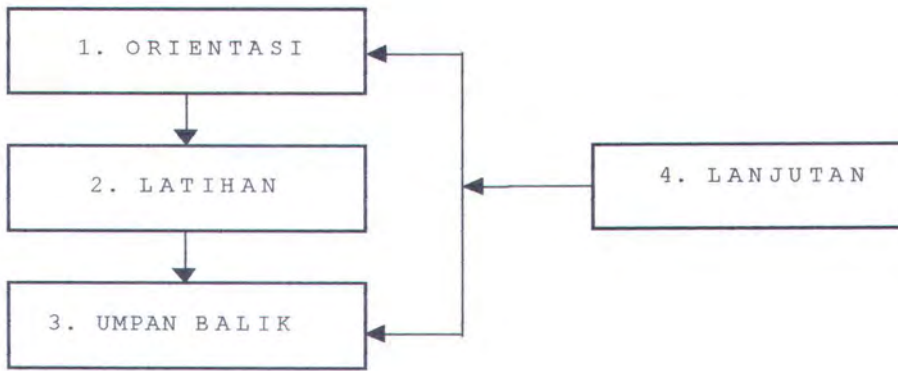
1. Siswa/mahasiswa akan mengerti konsep-konsep dasar dan ide-ide lebih baik.
2. Membantu dalam menggunakan ingatan dan transfer pada situasi-situasi proses belajar yang baru.
3. Mendorong siswa/mahasiswa untuk berfikir dan bekerja atas inisiatif sendiri.
4. Mendorong siswa/mahasiswa untuk berfikir intuitif dan merumuskan hipotesisnya sendiri.

5. Memberikan kepuasan yang bersifat intrinsik.
6. Situasi proses belajar menjadi lebih menarik.

Menurut teori pendidikan Gal'Perin [Utomo & Ruijter (1991)] proses belajar mengajar dapat digambarkan sebagai serangkaian empat tahap. Teori ini dianggap berlaku untuk proses belajar orang dewasa dibidang sains dan teknologi.

1. Mahasiswa berorientasi terhadap unsur-unsur ilmu yang penting, termasuk cara-cara penalaran yang khas untuk bidang itu.
2. Mahasiswa berlatih melakukan kegiatan-kegiatan bernalar itu, melalui kaitannya satu dengan yang lain.
3. Mahasiswa mendapat kesadaran tentang hasil belajar yang telah dicapai.
4. Mahasiswa melanjutkan proses belajar dengan cara orientasi - latihan - pemeriksaan.

Jadi dengan kata lain, menurut teori Gal'Perin suatu sasaran hanya akan tercapai bila mahasiswa berorientasi, berlatih dan melanjutkan proses belajar berdasarkan hasil umpan balik, seperti gambar berikut ini [Utomo & Ruijter (1991)]:



Gambar (2.1) Diagram proses belajar.

1. Orientasi:

Dasar teori yang diberikan kepada mahasiswa, dimana penyampaiannya tergantung tingkat kesulitan bahan yang disampaikan. Dengan orientasi ini mahasiswa akan dapat mengambil poin-poin yang mencakup pengertian dari teori yang diberikan, yang nantinya merupakan bekal baginya dalam menyelesaikan persoalan yang berkaitan.

2. Latihan:

Dari orientasi yang telah diberikan sebelumnya mahasiswa harus melakukan latihan-latihan untuk mendapatkan pengertian yang lebih jelas sekaligus menerapkan materi dan cara-cara ilmu dalam mengatasi masalah.

3. Umpan balik:

Setelah melakukan latihan-latihan mahasiswa akan diterangkan tentang kesalahan-kesalahan yang dilakukan dalam menyelesaikannya.

4. Lanjutan:

Memberikan dorongan dan kesempatan kepada mahasiswa untuk melanjutkan latihan-latihan.

Mengacu pada teori pendidikan diatas bahwa latihan perlu dilakukan bagi mahasiswa, selanjutnya perlu diadakan metode pengajaran yang selain memberikan latihan sekaligus juga memberikan motivasi kepada mahasiswa agar lebih giat melakukan latihan. Metode pengajaran tersebut salah satunya dengan memberikan kesempatan praktikum kepada mahasiswa. Dimana dengan praktikum tersebut diharapkan dapat mencapai tujuan-tujuan sebagai berikut [Rooijakkers (1990)]:

a. Tujuan Kognitif

Tujuan yang bersangkutan dengan pengetahuan dan pengertian:

- a. Melatih agar teori dapat dimengerti.
- b. Agar segi-segi teori yang berlainan dapat diintegrasikan.
- c. Agar teori dapat diterapkan pada keadaan problem yang nyata.

b. Tujuan Afektif

Tujuan yang berhubungan dengan mengubah minat, sikap, nilai dan alasan.

1. Mahasiswa dapat merencanakan kegiatan secara mandiri.
2. Terjalin kerjasama antar mahasiswa.

3. Mahasiswa dapat mengkomunikasikan secara timbal balik informasi yang mereka peroleh mengenai bidangnya.
4. Mahasiswa lebih menghargai bidangnya.

c. Tujuan Psikomotoris

Tujuan yang berkaitan dengan ketrampilan menggunakan tangan, mata, telinga dan alat indera lainnya.

1. Mahasiswa mampu memasang peralatan sehingga betul-betul berjalan.
2. Mahasiswa menguasai peralatan dan instrumen tertentu.

Selanjutnya agar tujuan-tujuan diatas dapat dicapai, praktikum yang disusun haruslah memiliki sifat-sifat sebagai berikut [Utomo & Ruijter (1991)]:

a. Kesahihan (Validity).

Praktikum adalah kegiatan ilmiah yang dilakukan untuk menguji kebenaran suatu teori, untuk itu metode praktikum yang digunakanpun harus sah dan dapat dipercaya.

b. Keterandalan (Realiability).

Dalam melakukan percobaan untuk mencapai kesahihan perlu dilakukan secara berulang-ulang dan hasil yang dicapai adalah sama. Disini bertujuan untuk meyakinkan orang lain akan kesahihan dari hasil praktikum yang telah dilaksanakan.



d. Keseksamaan (Accuracy) .

Keseksamaan dilakukan untuk mendapatkan hasil yang presisi sehingga perbedaan angka yang dihasilkan masih dalam batas yang telah ditentukan.

e. Kedayagunaan (Efficiency) .

Praktikum tidak akan berarti apa-apa bila tidak memberikan daya guna atau manfaat yang sebesar-besarnya baik bagi pelaku maupun orang lain.

II.3. TEORI ANALISA KEANDALAN DAN RESIKO

II.3.1. Konsep Probabilitas

Probabilitas suatu kejadian adalah suatu perkiraan terjadinya kejadian tersebut berdasarkan data yang ditentukan secara empiris maupun hasil kesimpulan-kesimpulan yang logis. Penerapan konsep probabilitas pada kehidupan nyata dapat dibedakan dalam 3 dasar pemikiran yaitu:

1. Probabilitas yang diperoleh dari hasil observasi kejadian serupa atau percobaan sebelumnya.
2. Probabilitas yang memenuhi aksioma-aksioma tertentu dan dengan alasan deduktif dari probabilitas suatu kejadian untuk menentukan probabilitas kejadian lainnya.

3. Prediksi fisis berdasarkan bilangan-bilangan probabilitas berdasarkan kejadian sampel untuk menentukan kemungkinan suatu kejadian.

Konsep probabilitas yang dikemukakan di sini adalah konsep yang berdasarkan aksioma-aksioma tertentu dan dengan hasil pemikiran deduktif mengenai probabilitas suatu kejadian untuk menentukan probabilitas kejadian selanjutnya.

Setiap kejadian mempunyai peluang terjadinya kejadian tersebut. Peluang terjadinya (*Probability of Occurrence*) dapat dimasukkan dalam range antara 0-1 [O'Connor (1991)]. Jika peluang terjadinya suatu kejadian sama dengan 0 artinya kejadian tersebut tidak akan pernah terjadi, sebaliknya bila peluang terjadinya sama dengan 1 maka kejadian tersebut pasti akan terjadi. Sebuah koin mempunyai peluang setengah untuk salah satu sisinya, dan sebuah dadu mempunyai peluang $1/6$ untuk tiap sisinya. Seperti pada koin dan dadu, kejadian-kejadian ini merupakan kejadian bebas, yaitu jika kejadian yang telah terjadi pada waktu lampau tidak mempengaruhi kejadian selanjutnya [O'Connor (1991)]. Seperti pada koin dan dadu, kemungkinan munculnya salah satu sisinya pada setiap pelemparannya tidak dipengaruhi oleh pelemparan sebelumnya.

Jika diasumsikan bahwa koin atau dadu adil/sempurna maka kita dapat memperkirakan peluang kejadian

munculnya salah satu sisi pada koin atau dadu, dan tiap sisi selalu mempunyai peluang yang sama. Pada banyak sistem, misalnya pengambilan sampling pada suatu tumpukan hasil produksi, kemungkinannya hanya dapat ditentukan dari hasil percobaan statistik sebelumnya.

Probabilitas dapat didefinisikan dalam dua cara [O'Connor (1991)]:

1. Jika sebuah kejadian dapat terjadi dalam N cara yang sama, dan jika kejadian A terjadi dalam n cara, maka probabilitas terjadinya A adalah:

$$P(A) = n/N \quad (2.1)$$

2. Jika dalam suatu percobaan, sebuah kejadian A terjadi n kali dari N percobaan dan N merupakan suatu nilai yang besar, probabilitas dari kejadian A mendekati n/N , yaitu:

$$P(A) = \lim_{n \rightarrow \infty} \left(\frac{n}{N} \right) \quad (2.2)$$

Definisi pertama mencakup contoh pada koin dan dadu dan definisi kedua lebih mengarah pada *quality control* dan *reliability*.

Suatu angka probabilitas mengarahkan kita pada suatu tingkat kepercayaan pada suatu kejadian yang akan berlangsung. Misalnya dalam suatu penentuan kekuatan suatu struktur A dan struktur B , bila peluang kegagalan untuk struktur A adalah 0.3 dan peluang kegagalan untuk struktur B adalah 0.4, maka kita akan lebih

berkeyakinan bahwa kekuatan struktur A lebih baik atau lebih handal dari pada struktur B.

Berikut notasi-notasi dasar dan hukum-hukum probabilitas [Walpole & Myers (1986)]:

1. Peluang terjadinya suatu kejadian (E) dinotasikan sebagai $P(E)$.

2. Peluang untuk semua kejadian:

$$0 \leq P(E) \leq 1 \quad (2.3)$$

3. Jika S adalah ruang sampel, maka:

$$P(S) = 1 \quad (2.4)$$

4. Peluang gabungan terjadinya E_1 dan E_2 dinotasikan

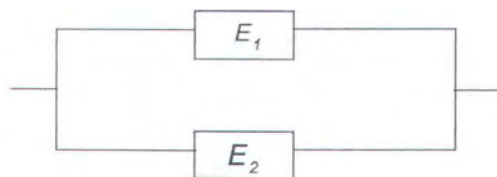
$$P(E_1 \cap E_2) \text{ atau } P(E_1 \times E_2).$$

5. Peluang terjadinya E_1 atau E_2 dinotasikan $P(E_1 \cup E_2)$ atau $P(E_1 + E_2)$

6. Peluang kondisional/bersyarat terjadinya E_1 , dengan syarat E_2 telah terjadi dinotasikan $P(E_1|E_2)$

5. Peluang komplemen, yaitu peluang tidak terjadinya E dinotasikan:

$$P(\bar{E}) = 1 - P(E) \quad (2.5)$$



gambar 2.3 Dual Redudant System [O'Connor (1991)]

6. Bila E_1 dan E_2 adalah dua kejadian sembarang yang saling bebas, maka:

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2) \quad (2.6)$$

$$P(A|B) = P(A|B) = P(A) \quad (2.7)$$

$$P(AB) = P(A) \cdot P(B) \quad (2.8)$$

$$P(B|A) = P(B|A) = P(B) \quad (2.9)$$

7. Bila E_1 dan E_2 adalah dua kejadian yang terpisah, maka:

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) \quad (2.10)$$

$$P(E_1 E_2) = P(E_1) \cdot P(E_2|E_1) = P(E_2) \cdot P(E_1|E_2) \quad (2.11)$$

8. Jika (dan hanya jika) kejadian $E_1, E_2 \dots E_n$ adalah *mutually exclusive*, yaitu kejadian satu tidak berhubungan dengan kejadian yang lain dalam ruang sampel maka:

$$P\left(\bigcup_{i=1}^n E_i\right) = \sum_{i=1}^n P(E_i) \quad (2.12)$$

9. Jika kejadian E_1 dan E_2 tak *mutually exclusive*, maka dipakai dalil De Morgan:

$$P(E_1 \cup E_2 \cup \dots \cup E_n) = 1 - \overline{P(E_1 \cap E_2 \cap \dots \cap E_n)} \quad (2.13)$$

10. Probabilitas bersyarat, yaitu probabilitas dimana kejadian bergantung pada terjadinya (tak terjadinya) kejadian lain

$$P(E_1|E_2) = \frac{P(E_1) \cdot P(E_2)}{P(E_2)} \quad (2.14)$$

II.3.2. Konsep Keandalan

II.3.2.1. Konsep ketidakpastian.

Dalam setiap proses perancangan sistem rekayasa akan ditemui ketidakpastian-ketidakpastian. Adanya ketidakpastian tersebut dikarenakan oleh keterbatasan pengetahuan manusia tentang alam dan adanya ketidakseragaman variabel-variabel yang terdapat pada sistem rekayasa tersebut. Secara umum ketidakpastian-ketidakpastian dalam proses perancangan dikelompokkan menjadi 3 bagian, yaitu:

1. Ketidakpastian fisik

Adanya ketidakseragaman beban, sifat-sifat material dan ukuran material.

2. Ketidakpastian statistik

Ketidakpastian yang diakibatkan ketidakaksamaan kualitas dan kuantitas sampel data.

3. Ketidakpastian model

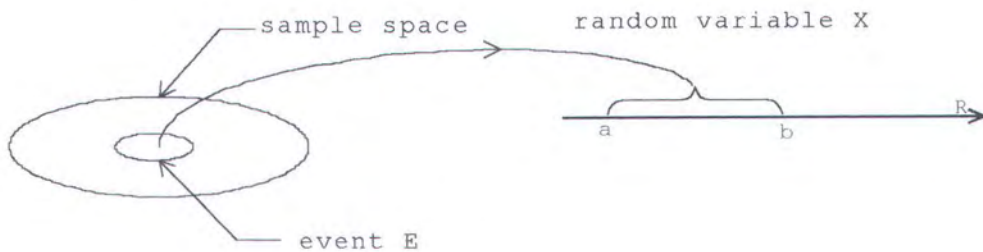
Ketidakaksamaan antara sistem nyata dengan model yang dipakai untuk mewakili sifat-sifat sistem tersebut.

Analisa keandalan menitik beratkan pada perlakuan dan pemilihan ketidakpastian secara rasional dan menyangkut masalah pengambilan keputusan yang rasional. Perlakuan rasional dari ketidakpastian bisa dinyatakan sebagai berikut [Thoft-Christensen & Baker (1982)]:

"Semua besaran (kecuali konstanta fisik dan matematik) yang langsung termasuk dalam perhitungan teknik, dalam kenyataannya tidak tentu. Kenyataan ketidakpastian ini secara implisit disebutkan dalam kode-kode, yaitu apa yang disebut "faktor keamanan".

II.3.2.2. Perubah Acak dan Fungsi Kerapatan Peluang

Perubah acak adalah suatu fungsi dari sebuah kejadian dalam satu ruang sampel yang dimasukkan dalam suatu garis R , seperti pada gambar berikut [Thoft-Christensen & Baker (1982)]:



Gambar 2.2 Perubah acak

Jika dalam ruang sampel tersebut diskret maka disebut perubah acak diskret dan bila dalam ruang sampel tersebut kontinyu maka disebut perubah acak kontinyu.

Bila kita ingin mengukur probabilitas kejadian P dari beberapa kejadian perubah acak maka digunakan analisa perhitungan dan grafik. Untuk fungsi diskret, fungsi probabilitasnya didefinisikan:

$$P_X(x) = P(X = x) \quad (2.15)$$

dimana, X =perubah acak

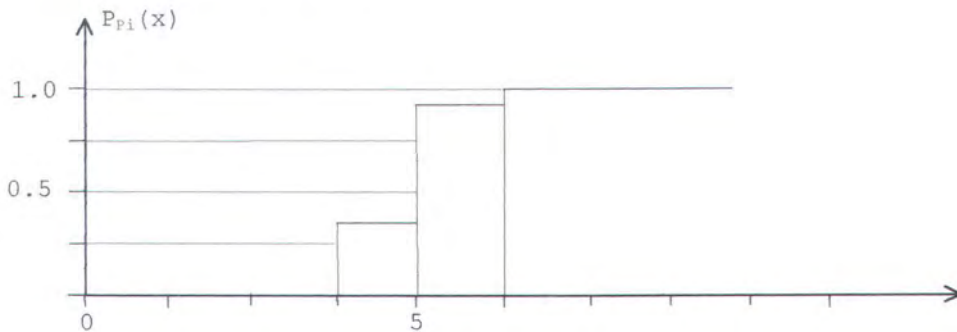
$$x = X_1, X_2, \dots, X_n$$

Sehingga *probability distribution function* (PDF):

$$P_X(X) = P(X \leq x) = \sum_{x_i \leq x} P_X(x_i) \quad (2.16)$$

Arti fisik persamaan diatas adalah probabilitas dari suatu kejadian perubah acak X yang sama atau kurang dari x .

Contoh grafik untuk fungsi diskret ini [Thoft-Christensen & Baker (1982)]:



Gambar 2.3 Grafik fungsi diskret

Untuk fungsi kontinyu, *probability distribution function* didefinisikan:

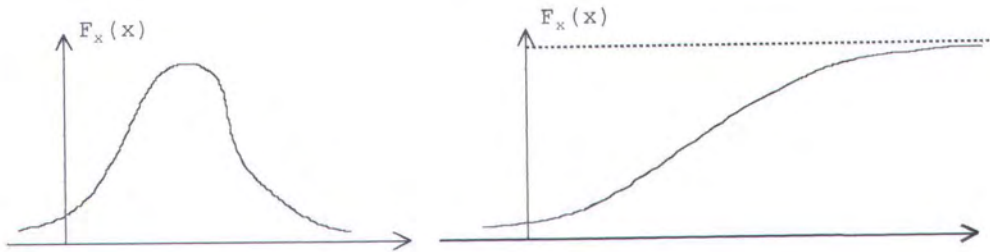
$$F_X(x) = P(X \leq x) \quad (2.17)$$

Persamaan ini dapat dianggap sebagai *cumulative distribution function* (CDF) yang sering digunakan untuk menurunkan fungsi probabilitas yang disebut *probability density function* (PDF):

$$f_x(X) = \frac{df_x(X)}{dX} \quad (2.18)$$

$$F_x(X) = P(X \leq x) = \int_{-\infty}^x f_x(X) dX \quad (2.19)$$

Bila interval waktu $dx \cong 0$ maka diagram histogram tersebut akan berubah menjadi suatu kurva yang menerus [Thoft-Christensen & Baker (1982)], misalnya:



Gambar 2.4 Kurva PDF dan CDF

Dari gambar diatas didapatkan:

$$F_x(x) = 0, \text{ bila } x = -\infty$$

$$F_x(x) = 1, \text{ bila } x = \infty$$

Dari persamaan (II.2.1.5) $F_x(X) = \int_{-\infty}^x f_x(X) dX$, maka:

$$\int_{-\infty}^x f_x(X) dX = 1 \quad (2.20)$$

Selanjutnya untuk menentukan mean (m), standard deviasi (σ), varian (VAR) dan covarian (COV) adalah:

$$\mu = \int_{-\infty}^{\infty} x \cdot f_x(X) dx \quad (2.21)$$

$$\text{VAR}(x) = \int_{-\infty}^{\infty} (x - \mu)^2 \cdot f_x dx \quad (2.22)$$

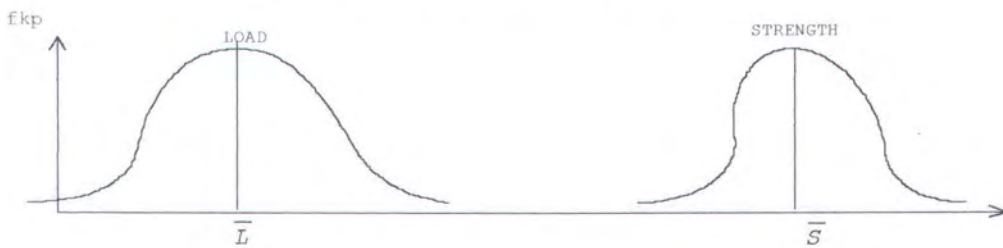
$$\sigma = \sqrt{\text{VAR}} \quad (2.23)$$



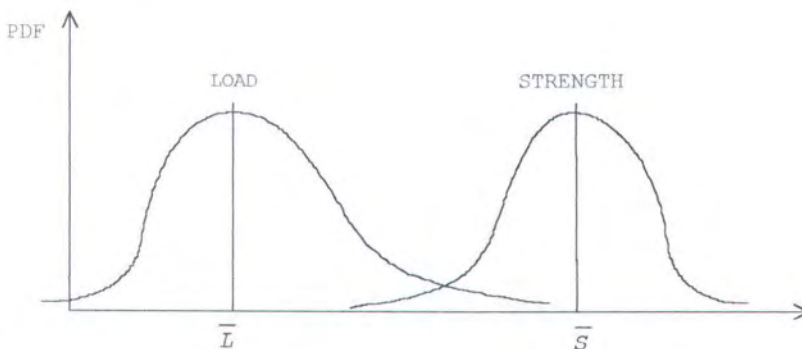
$$COV = \frac{\sigma_X}{\mu_X} \quad (2.24)$$

II.3.2.3. Diagram Interferensi

Apabila suatu sistem diasumsikan memiliki kekuatan yang terdistribusi dengan fungsi kerapatan peluang (f_{kp_s}), dan efek pembebanannya terdistribusi dengan fungsi f_{kp_l} . Apabila kedua fkp tersebut terpisah sama sekali maka dapat dikatakan bahwa sistem tersebut memiliki keandalan $K = 1$, atau tidak pernah gagal.



Gambar (2.5) Distribusi beban dan kekuatan yang saling lepas



Gambar (2.6) Distribusi beban dan kekuatan yang saling memotong

Pada kedua gambar (2.5) dan (2.6) diatas [O'connor (1991)] setiap distribusinya mempunyai nilai rata-rata (mean) dengan notasi $\bar{L}(\mu_L)$ dan $\bar{S}(\mu_S)$ sedangkan nilai

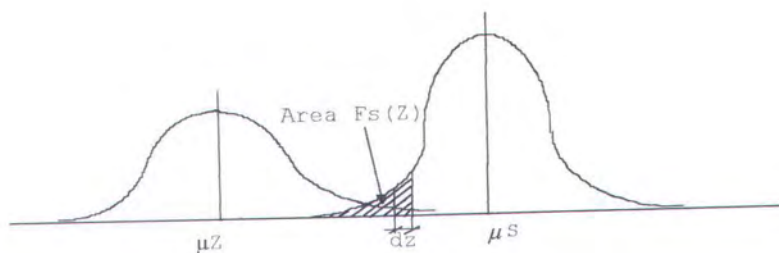
simpangan bakunya (*standard deviasi*) dinotasikan σ_L dan σ_S . Seperti telah dijelaskan diatas bila kedua distribusinya saling lepas maka sistem dikatakan tidak pernah gagal (gambar(2.5)). Sebaliknya bila kedua distribusinya saling memotong (*overlap*) maka sistem akan mengalami kegagalan gambar (2.6)).

Dari distribusi kekuatan dan pembebanan diatas dapat ditentukan dua faktor [O'connor (1991)]:

$$\text{Safety Factor (SM)} = \frac{\mu_S - \mu_L}{\sqrt{(\sigma_S^2 + \sigma_L^2)}} \quad (2.25)$$

$$\text{dan Loading Roughness (LR)} = \frac{\sigma_L}{\sqrt{(\sigma_S^2 + \sigma_L^2)}} \quad (2.26)$$

Sebuah contoh yang sederhana, pada gambar (2.7) memperlihatkan fungsi kerapatan peluang (*Probability Density Function; pdf*) untuk variabel beban (Z) dan kekuatan. (S)



gambar 2.7 Interferensi dua distribusi peluang (Beban-Kekuatan)

Kita tetapkan suatu fungsi sederhana $g(s,z)$ yang disebut fungsi keadaan batas, yang menerangkan batas aman (*safety margin*) M antara kekuatan dan beban yang bekerja, yaitu:

$$M = g(s, z) = S - Z$$

S dan Z adalah variabel acak dan dapat diasumsikan mempunyai distribusi nilai. Beberapa keadaan berikut menerangkan kemungkinan yang dapat terjadi pada beam tersebut:

$$(i) M = g(s, z) < 0$$

Keadaan dimana beam mengalami kegagalan, artinya beban Z melebihi kekuatan S

$$(ii) M = g(s, z) > 0$$

Keadaan aman, artinya beban Z lebih kecil dari kekuatan S.

$$(iii) M = g(s, z) = 0$$

Keadaan dimana batas garis antara keadaan aman dan gagal.

Peluang terjadinya kegagalan pada keadaan pertama (i), dapat diterangkan secara sederhana sebagai berikut:

Pada gambar (2.7) diatas, menunjukkan bahwa peluang harga beban Z yang berada pada interval dz , $(z-dz/2 ; z+dz/2)$ adalah $f_s(z)dz$. Kegagalan akan terjadi jika harga kekuatan S lebih kecil dari harga Z dan peluang kejadian ini adalah $\int_0^Z f_s(z)dz$ atau hasil fungsi

distribusi kumulatif (*Cumulative Distribution Function*; CDF) kekuatan beam pada nilai Z, $F_s(Z)$. Dengan memakai

persamaan (2.8) maka peluang kejadiannya adalah $F_S(Z)f_Z(z)dz$, untuk total peluang kejadian P_f adalah:

$$P_f = \int_0^{\infty} F_S(Z)f_Z(z)dz \quad (2.27)$$

II.3.2.4. Distribusi Peluang

Distribusi Peluang yang disajikan dalam penulisan Tugas Akhir ini hanya Distribusi Normal dan Log normal karena kedua distribusi tersebut yang paling sering ditemui di lapangan.

1. Distribusi Normal (Gaussian)

Persamaan distribusi normal ditulis sebagai berikut:

$$f(x) = \frac{1}{\sqrt{\sigma(2\pi)}} \exp\left[-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right] \quad (2.28)$$

dimana m adalah rata-rata (*mean*), s adalah *standar deviasi*. Populasi yang mengikuti distribusi ini mempunyai variasi yang bentuknya sama yang ditentukan oleh nilai rata-ratanya (i.e. nilai kecondongannya = 0). Karena ekor dari distribusi normal adalah simetris maka lebar ekor sebelah kiri sama dengan sebelah kanannya.

Sebuah alasan yang penting sehingga distribusi ini diaplikasikan secara luas adalah ketika beberapa distribusi saling ber-*interferensi* tanpa mempertimbangkan bagaimana bentuk distribusinya

maka hasil peluang distribusi gabungan dapat didekati dengan distribusi normal. Distribusi normal ini sangat cocok untuk *quality control* dan beberapa observasi keandalan seperti ukuran material dan umur suatu bagian sistem.

Standartisasi fungsi distribusi (i.e. $\mu=0$ dan $\sigma=1$):

$$z = (x - \mu) / \sigma \quad (2.29)$$

sehingga persamaan 2.27 dapat ditulis:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} z^2\right) \quad (2.30)$$

2. Distribusi Log normal

Persamaan distribusi Log normal ditulis sebagai berikut:

$$f(x) = \left\{ \frac{1}{\sqrt{\sigma x}(2\pi)} \exp\left[-\frac{1}{2} \left(\frac{\ln x - \mu}{\sigma}\right)^2\right] \right\}_{(x < 0)}^{(x \geq 0)} \quad (2.31)$$

Distribusi ini mempunyai daerah yang tajam sehingga sering lebih cocok untuk data keandalan. Misalnya, populasi dengan karakteristik tertentu. Distribusi ini muncul sebagai distribusi terbatas pada saat sebuah variabel acak dihasilkan dari beberapa variabel acak yang bebas dan terdistribusi secara identik [Ship Structure Committee (1990)]. Distribusi ini dapat dikatakan distribusi normal dengan variabel $\ln x$. Mean dan Standard Deviasi-nya adalah:

$$\text{Mean} = \exp(\mu + \sigma^2/2) \quad (2.32)$$

$$SD = [\exp(2\mu + 2\sigma^2) - \exp(2\mu + \sigma^2)]^{0.5} \quad (2.33)$$

dimana μ dan σ adalah Mean dan Standard Deviasi dalam data \ln .

Jika, $\mu \gg \sigma$, Distribusi Log normal akan mendekati Distribusi Normal.

II.3.2.5. Metode Mean Value First Order Second Moment (MVFOSM)

Dalam metode MVFOSM ini, masukan yang diperlukan dalam perhitungan adalah harga rata-rata (*mean values* atau *first moment*) dan simpangan baku (*standard deviation* atau *second moment*) dari perubah-perubah acak tersebut. Jadi distribusi atau fungsi kerapatan peluang fkp perubah-perubah tersebut tidak diperlukan secara langsung.

Suatu sistem dikatakan gagal jika kekuatan (*resistance*) sistem R kurang dari beban yang diterimanya Q (*load*). Dalam teori keandalan perbedaan antara R dan Q dinamakan fungsi keandalan (*reliability function*) Z ,

$$Z = R - Q \quad (2.34)$$

Jika diasumsikan R dan Q adalah variabel random independen dan mempunyai distribusi normal yang diketahui harga rata-rata dan simpangan bakunya, maka

fungsi keandalan Z yang terdistribusi normal juga dapat dicari harga rata-rata dan simpangan bakunya:

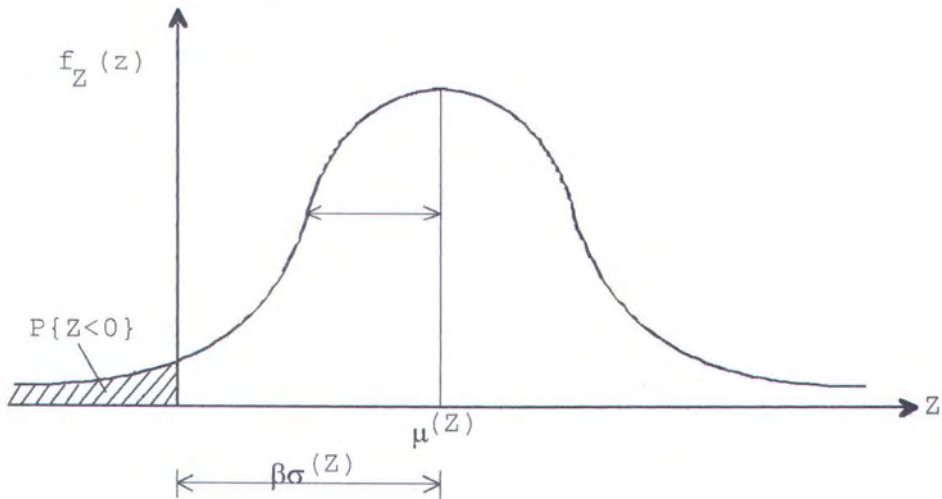
$$\overline{Z} \approx \overline{R} - \overline{Q}, \quad (2.35)$$

$$\sigma^2_Z = \sigma^2_R + \sigma^2_Q \quad (2.36)$$

$$\sigma^2_R \approx \sum \left[\frac{\partial R}{\partial x_i} \sigma_i \right]_{x_i = \overline{x_i}}^2 \quad (2.37)$$

$$V = \frac{\sigma}{R} \quad (2.38)$$

$$V^2_R \approx \sum \left[\frac{\partial R}{\partial x_i} \frac{x_i}{R} \right]_{x_i = \overline{x_i}}^2 \cdot V^2_{x_i} \quad (2.39)$$



Gambar 2.12 Fungsi kerapatan peluang untuk $Z=R-Q$

Probabilitas kegagalan dari gambar di atas [Vrouwenvelder, Karadeniz & Van Manen] dapat dihitung dari:

$$P(Z < 0) = \int_{-\infty}^0 f_Z(z) dz = \Phi_N(-\beta) \quad (2.40)$$

$$\beta = \frac{\mu_Z}{\sigma_Z} \quad (2.41)$$

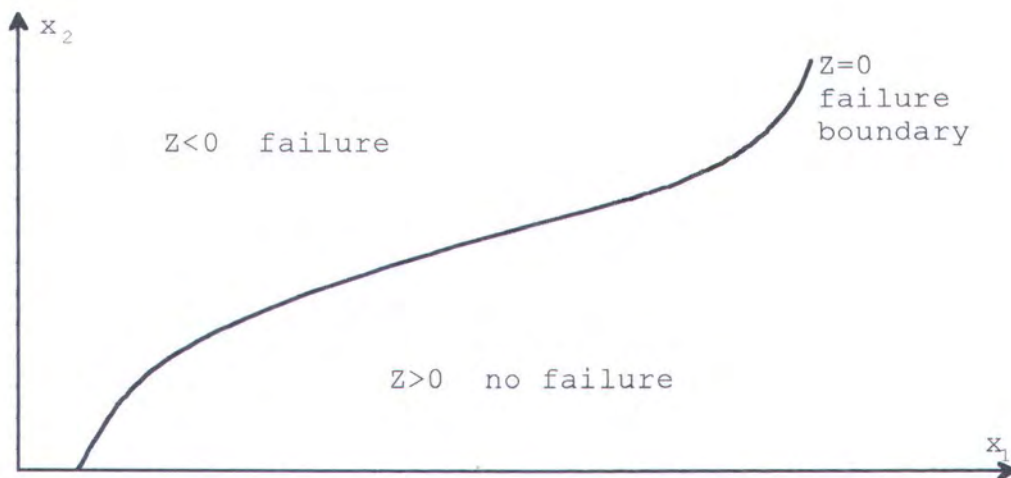
dimana:

$f_Z(z)$ = probability density function dari Z

$\Phi_N(\cdot)$ = fungsi distribusi normal standar

β = reliability index

Bila fungsi keandalan dengan n variabel stokastik, misalnya $x_1, x_2, x_3, \dots, x_n$ dan diasumsikan x_i adalah *mutually independent* dengan nilai rata-rata dan simpangan bakunya diketahui, maka fungsi tersebut dikatakan gagal jika berharga negatif dan sebaliknya, seperti gambar berikut [Vrouwenvelder, Karadeniz & Van Manen]:



Gambar 2.13 Definisi batas kegagalan

titik-titik dengan harga $Z=0$ disebut batas bidang gagal (*failure boundary*). Dengan memakai formula seri Taylor orde pertama beberapa linierisasi titik x_i^* , fungsi Z menjadi:

$$z = z(x_1^*, \dots, x_n^*) + \sum_{i=1}^n (x_i - x_i^*) \frac{\partial z}{\partial x_i} \quad (2.42)$$

$\partial z / \partial x_i$ adalah *derivatif parsial* yang dievaluasi pada titik x_i^* . Maka rata-rata dan varian z adalah:

$$\mu(z) = z(x_1^*, \dots, x_n^*) + \sum_{i=1}^n (x_i - x_i^*) \frac{\partial z}{\partial x_i} \quad (2.43)$$

$$\sigma^2(z) = \sum_{i=1}^n \left[\sigma(x_i) \frac{\partial z}{\partial x_i} \right]^2 \quad (2.44)$$

Maka probabilitas kegagalannya:

$$P(z < 0) = \Phi_N(-\beta) \quad (2.45)$$

dimana, $\beta = \frac{\mu(z)}{\sigma(z)}$

Metode *MVFOSM* ini mempunyai tiga kekurangan [Ship Structure Committee (1990)]:

Pertama, jika fungsi batas $g(\cdot)$ tidak linier dan linierisasi mengambil tempat pada nilai *mean* dari x_i , kesalahan mulai terjadi pada pertambahan jarak dari titik linierisasi karena pengabaian orde yang lebih tinggi.

Kedua, metode ini tidak dapat menghasilkan nilai yang sama untuk pengambilan fungsi batas yang berbeda pada persoalan yang sama. Jadi indeks keamanan β tergantung pada bagaimana formulasi fungsi keadaan batas.

Ketiga, pada metode *MVFOSM*, indeks keandalan β dapat dihubungkan dengan probabilitas

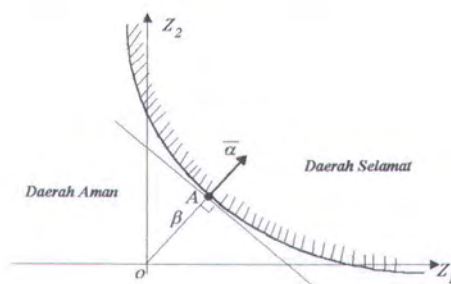
kegagalan jika variabel x_i terdistribusi normal (dan fungsi batas $g(.)$ adalah linier dalam x_i). Kita tahu bahwa kebanyakan variabel acak dasar mengikuti distribusi Weibull atau distribusi Eksponensial, sehingga butuh normalisasi distribusi.

II.3.2.6. Indeks Keandalan Hashofer dan Lind.

Kekurangan pertama dan kedua metode MVFOSM dapat dihindari dengan menggunakan cara yang diberikan Hashofer dan Lind. Prosedur yang dikembangkan oleh Hasofer dan Lind yaitu variabel x_i (variabel pada fungsi batas) ditransformasikan ke distribusi normal standar:

$$y_i = \frac{x_i - \bar{x}_i}{\sigma x_i} \quad (2.46)$$

Indeks keandalan Hasofer dan Lind didefinisikan sebagai jarak terpendek dari titik pusat 0 ke permukaan bidang gagal pada sistem koordinat variabel-variabel normal standar. Pada gambar (2.8), indeks keandalan sama dengan jarak OA. Titik A disebut design point.



Gambar 2.9 Daerah batas

Pada gambar (2.9) [Thoft-Christensen & Murotsu (1986)] vektor α adalah vektor unit normal pada titik perancangan A, dimana vektor α mempunyai arah tegak lurus dengan bidang A. Pada beberapa kasus berdimensi-n

$$\bar{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n) \quad (2.47)$$

dan jarak β dapat dicari dengan menggunakan iterasi persamaan n+1

$$f(\beta\alpha_1, \beta\alpha_2, \dots, \beta\alpha_n) = 0 \quad (2.48)$$

Untuk memperoleh titik A dapat dilakukan dengan menyelesaikan persamaan-persamaan berikut:

$$\alpha_i = -\frac{1}{k} \frac{\delta f}{\delta z_i} (\beta\bar{\alpha}), \quad i=1, \dots, n \quad (2.49)$$

dimana bidang kegagalan diekspresikan sebagai:

$$f(\bar{z}) = f(z_1, z_2, \dots, z_n) = 0 \quad (2.50)$$

$$\bar{z}_i = \beta\alpha_i \quad (2.51)$$

dan

$$k = \sqrt{\sum_{i=1}^n \left(\frac{\delta f}{\delta z_i} (\beta\bar{\alpha}) \right)^2} \quad (2.52)$$

Prosedur ini yang diajukan oleh Hasofer dan Lind disebut *Advanced First Order Second Moment (AFOSM) method*.

Prosedur AFOSM dapat diringkas sebagai berikut:

1. Tentukan *failure function* untuk moda kegagalan yang ditinjau.



2. Karakterisasi perubah-perubah dasarnya, x_i .
3. Transformasikan setiap perubah dasar menjadi perubah dasar baku, y_i^*
4. Definisikan sebuah vektor satuan α_i^* dengan $z_i^* = \beta \alpha_i^*$.
5. Untuk persamaan-persamaan (2.49) sampai (2.50):
 - a. Tentukan $\alpha_1, \alpha_2, \dots, \alpha_n$ percobaan dan β percobaan.
 - b. Hitung α_i dan β yang baru sampai konvergen.

II.3.2.7. Simulasi Monte Carlo.

Metode AFOSM akan berfungsi baik apabila fungsi kinerja $FK(X)$ linier atau sedikit non-linier. Apabila $FK(X)$ sangat non linier, maka metode Simulasi Monte Carlo dapat merupakan alternatif metode yang lebih bermanfaat. Hal ini terjadi karena pada metode Simulasi Monte Carlo tidak melakukan diferensiasi parsial seperti pada metode AFOSM, melainkan dengan menggunakan sebuah *random number generator*.

Nama Metode Monte Carlo diperkenalkan pertama kali pada tahun 1944 oleh Von Newmann, Ulam dan Fermi sebagai kode dari misi rahasia mereka dalam proyek difusi netron untuk mengembangkan bom atom pada Perang Dunia II di Laboratorium Los Alamos [Watson & Blackstone (1981)]. Nama ini dipilih sesuai nama kota Monte Carlo yang terkenal dengan *Roulette* yang

merupakan salah satu peralatan sederhana yang dapat digunakan sebagai *generating random number*.

Simulasi Monte Carlo digunakan secara umum pada penyelesaian kasus-kasus yang menggunakan variabel acak atau yang biasa diasumsikan sebagai distribusi peluang [Ship Structure Committee (1990)]. Simulasi ini menggunakan teknik sampel statistik, yaitu sekumpulan angka acak yang dicari dengan *random number generator*. Anggap persamaan fungsi keadaan batas adalah $g(x) = g(x_1, x_2, \dots, x_n)$, dimana x_i adalah variabel acak yang mewakili variabel acak dasar. Selanjutnya setiap angka acak x_i dimasukkan dalam persamaan fungsi $g(x)$, hasil dari setiap harga x_i yang dimasukkan dibandingkan dengan nol. Bila $g(x)$ bernilai negatif maka dihitung sebagai kegagalan, sebaliknya bila $g(x)$ positif dihitung sebagai keadaan aman. Proses ini dilakukan berulang-ulang sebanyak jumlah angka acak yang diinginkan. Nilai keandalan dari simulasi ini adalah jumlah kegagalan dibagi dengan jumlah percobaan yang dilakukan atau jumlah angka acak yang dibuat [Rosyid (1996)]. Atau dapat diekspresikan dalam suatu formulasi sebagai berikut [Ship Structure Committee (1990)]:

$$p_f = P[g(x) \leq 0] = \lim_{N \rightarrow \infty} \frac{n}{N}$$

dimana: N adalah jumlah percobaan,

n adalah jumlah kegagalan

Hasil dari Simulasi Monte Carlo sangat tergantung dari jumlah sampel acak yang diambil. Semakin banyak sampel acak yang diambil maka hasilnya akan semakin akurat. Selain itu juga sangat dipengaruhi apakah sampel acak yang diambil tersebut benar-benar acak dan berada pada batasan-batasan yang telah ditentukan sebelumnya.

Langkah pertama pada Metode Simulasi Monte Carlo ini adalah mentransformasikan angka acak yang dikeluarkan oleh *random number generator* (RNG) menjadi besaran fisis yang sesuai dengan fungsi kerapatan peluangnya. Hal ini disebabkan karena angka acak yang dikeluarkan oleh RNG memiliki fkp *uniform*, sedangkan perubah acak dalam bidang batas $g(x)$ seringkali tidak demikian (misalkan terdistribusi secara normal, lognormal, dsb.).

Angka acak yang dikeluarkan oleh komputer (*randomize*) biasanya merupakan *prosedure/function built-in* dari suatu bahasa program yang hampir dapat dipastikan berbentuk *linear congruential (uniform)* karena umumnya didapat dengan perhitungan rekursi dan siklus yang berulang-ulang, sehingga angka yang dihasilkan tidak benar-benar acak. Suatu angka yang sama akan keluar lagi pada putaran yang sangat lama, dan setiap angka yang keluar mempunyai peluang untuk keluar yang sama besar satu dengan yang lainnya.

Metode *Congruential* pertama kali diperkenalkan oleh D. H. Lehmer pada tahun 1949 [Watson & Blackstone (1981)]. Metode RNG ini mengeluarkan suatu deretan bilangan cacah (integer) $x_1, x_2, x_3, \dots, x_m$ dimana m adalah sebuah bilangan yang sangat besar sekali.

$$x_i = a x_{i-1} + b \pmod{m} \quad (2.53)$$

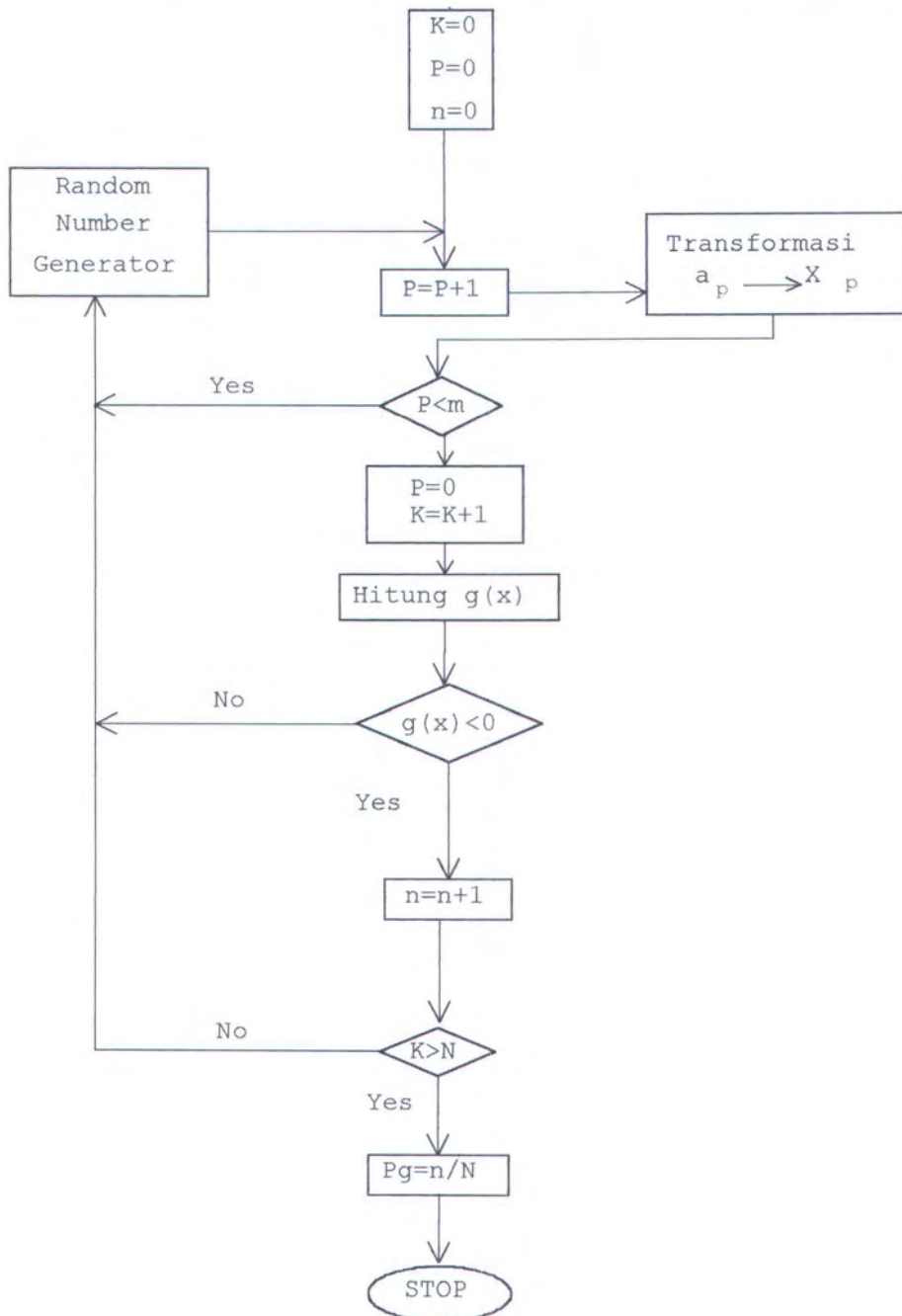
dimana a , b dan m bilangan integer positif dan hasil bagi X_i/m merupakan deret angka *pseduo-random* [Ship Structure Committee (1990)]. Seperti telah disinggung di atas deretan angka *pseduo-random* tersebut akan terulang kembali setelah sekian kali putaran. Jika pemilihan a , b dan x dilakukan secara tepat, maka periode ulang yang terjadi akan memiliki panjang maksimum, yaitu m . Dalam hal ini semua bilangan cacah (integer) antara 1 dan $m-1$ akan muncul, sehingga setiap pemilihan angka acak akan menghasilkan deret yang sama baiknya.

Sejumlah angka acak a_p , $p=1,2,\dots,m$, dengan fkp *uniform* yang dikeluarkan oleh RNG kemudian ditransformasikan menjadi x_p , $p=1,2,\dots,m$. Transformasi angka acak a_p menjadi X_p dapat dilakukan sesuai prosedur berikut [Rosyid (1996)]:

- a. X_p dari fungsi keadaan batas telah diketahui fkp-nya, rentang x_p tersebut kemudian dibagi menjadi i interval yang sama sepanjang dx .

- b. Setiap luas pias interval tersebut dihitung luasnya (anggap P_i), selanjutnya setiap a_p yang dikeluarkan RNG dibandingkan dengan P_i . Bila $P_i \leq a_p \leq P_{i+1}$, maka $x_1 = a_p$.

Berikut diagram alir program Simulasi Monte Carlo:



Gambar 2.10 Diagram alir simulasi monte carlo

II.3.2.8. Fault Tree Analysis

Fault Tree Analysis (FTA) adalah suatu metodologi deduktif untuk menentukan penyebab potensial dari kejadian, atau kegagalan sistem yang lebih umum selanjutnya memperkirakan probabilitas kegagalan tersebut. Dalam pengertian yang lebih sempit, FTA dapat dilihat sebagai suatu alternatif dari penggunaan *reliability block diagram*, yaitu suatu metode untuk menentukan kehandalan sistem yang menggunakan pendekatan tiap komponen atau sub-sistem dalam blok-blok yang kemudian digabung menjadi suatu sistem yang utuh. Tetapi FTA berbeda dengan RBD baik dalam pendekatan masalah maupun dalam ruang lingkup analisisnya.

Reliability Block Diagram lebih berorientasi pada suksesnya komponen-komponen (*success-oriented*), yaitu semua kegagalan yang disatukan untuk mendapatkan probabilitas bahwa suatu sistem akan gagal. Jadi studi keandalannya ditekankan pada penentuan tingkat keandalan (yaitu probabilitas bahwa sistem tidak akan gagal). Sebaliknya FTA hanya memperhatikan kejadian khusus yang tidak diharapkan (yaitu kegagalan yang membawa pada keadaan yang membahayakan keselamatan) dan menghitung probabilitas bahaya itu akan terjadi. Sedangkan kegagalan-kegagalan yang tidak menyebabkan bahaya pada keselamatan *Top Event* diabaikan. FTA

selanjutnya memecah sistem ke dalam detail yang terus bertambah untuk menentukan akar penyebab atau kombinasi dari penyebab-penyebab terjadinya *top event*. *Top event* biasanya adalah kegagalan yang membawa konsekuensi paling besar yang menimbulkan bahaya serius bagi keselamatan atau kerugian ekonomis yang besar.

Fault tree analysis dapat dipermudah dengan mengklasifikasikan kegagalan dalam tiga hal [Ramakumar (1993)]:

a. Primary failure

Adalah kegagalan yang terjadi bila sebuah komponen berfungsi dalam sebuah parameter operasi yang telah direncanakan.

b. Secondary failure

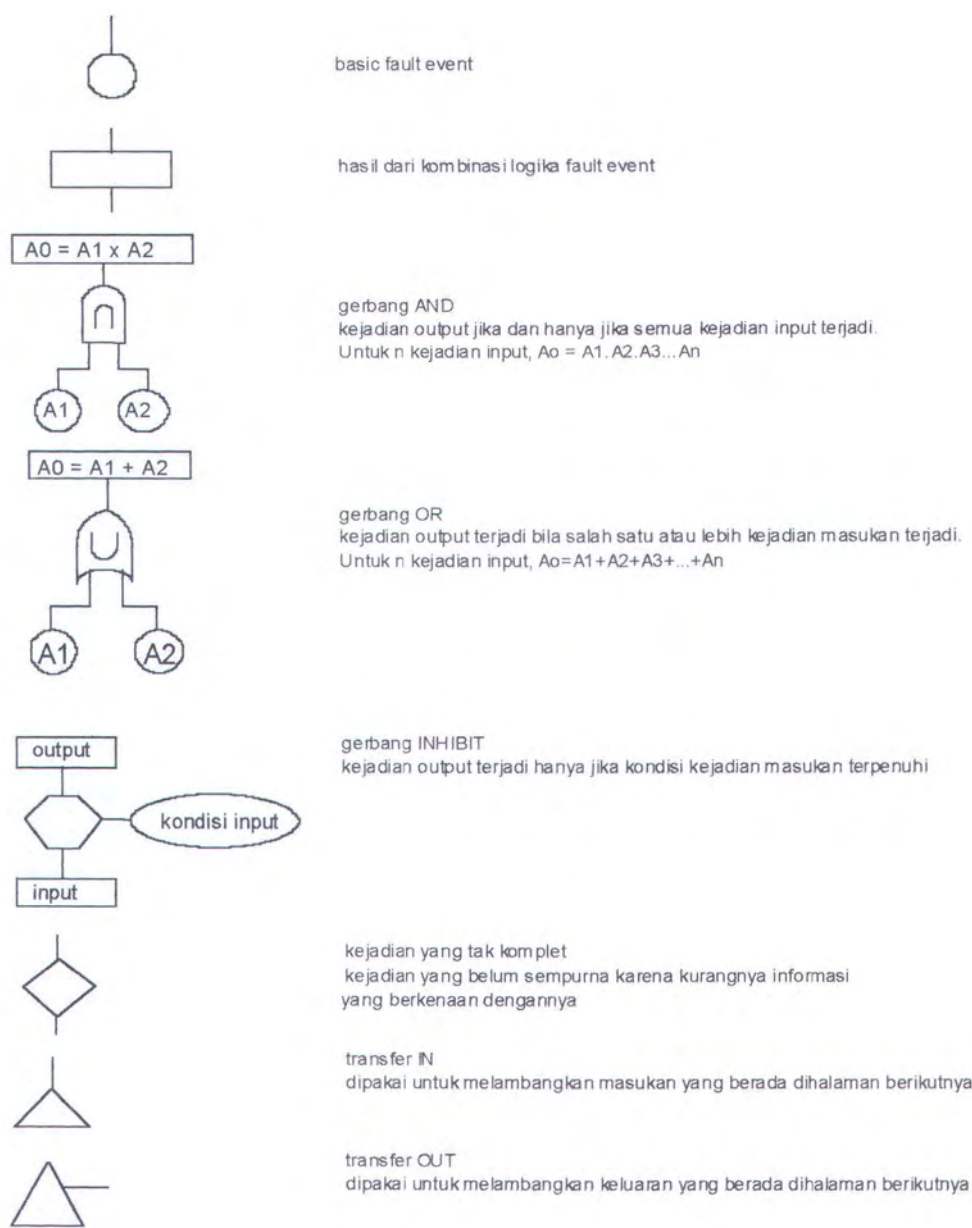
Secondary failure adalah kegagalan yang terjadi akibat berlebihan tegangan operasi atau lingkungan yang bekerja terhadap komponen tersebut.

c. Command failure

Command failure adalah kegagalan yang terjadi akibat kesalahan waktu atau tempat dalam pengoperasiannya.

Sebuah *fault tree* terdiri dari event yang dinyatakan dalam kotak dan gerbang (*gate*) logika. Terdapat dua jenis gerbang logika, yaitu gerbang OR dan AND. Gerbang OR digunakan untuk menunjukkan bahwa

kejadian hasil (*output event*) terjadi jika salah satu atau lebih dari kejadian masukan (*input event*) terjadi. Gerbang AND digunakan untuk menunjukkan kejadian hasil terjadi jika dan hanya jika semua kejadian masukan terjadi. Berikut adalah lambang-lambang yang digunakan dalam analisa *fault tree* [Ramakumar (1993)].



Gambar 2.11 Lambang-lambang dari metode FTA

Gerbang *inhibit* adalah kasus khusus untuk untuk gerbang *AND*. Keluaran hanya bila syarat kondisional untuk kejadian masukan terpenuhi.

FTA merupakan gabungan analisa kualitatif dan kuantitatif sekaligus. Pada analisa kualitatif, *fault tree* merupakan ekspresi logika yang memberikan *top event* sebagai kombinasi dari kejadian-kejadian kegagalan utama. Sedangkan pada evaluasi kuantitatif probabilitas munculnya *top event* diperiksa dalam probabilitas kejadian-kejadian utama.

Konstruksi dari *fault tree* itu sendiri memungkinkan analisa untuk mendapatkan pengertian yang lebih baik terhadap sumber-sumber kegagalan dan menjadi alat untuk memikirkan kembali desain dan operasi dari sistem guna memperkecil bahaya-bahaya potensial. Struktur *fault tree* dapat dianalisa untuk menentukan apakah kombinasi kegagalan komponen, kesalahan operasional, atau kegagalan-kegagalan lainnya dapat menyebabkan terjadinya *top event*.



BAB III

PENGEMBANGAN METODE PRAKTIKUM

Dialah Tuhan yang menjadikan kamu dapat berjalan didaratan, (berlayar) dilautan, sehingga apabila kamu berada didalam bahtera dan meluncurlah bahtera itu membawa orang-orang yang ada didalamnya dengan tiupan angin yang baik, dan mereka bergembira karenanya, (tiba-tiba) datanglah angin badai dan gelombang dari segenap penjuru menimpanya, mereka yakin bahwa mereka telah terkepung (bahaya), lalu mereka berdoa kepada Allah dengan mengikhlaskan ketaatan kepada-Nya semata-mata (seraya berkata):

“Sesungguhnya jika Engkau menyelamatkan kami dari bahaya ini, pasti kami akan termasuk orang-orang yang bersyukur”. (QS Yunus: 22)

BAB III

PENGEMBANGAN METODE PRAKTIKUM

III.1. PENENTUAN JENIS PRAKTIKUM

Mata kuliah analisa keandalan dan resiko adalah salah satu mata kuliah yang sangat penting dalam menunjang suatu proses perencanaan sistem-sistem rekayasa. Mata kuliah ini juga berhubungan dan mendukung beberapa mata kuliah lain dalam proses perencanaan tersebut. Selanjutnya mata kuliah ini diharapkan mampu memberikan bekal yang cukup bagi mahasiswa sebagai calon perancang. Dalam usaha memaksimalkan tujuan-tujuan mata kuliah ini seperti yang telah dijabarkan pada bab sebelumnya, diperlukan sarana penunjang yang dapat memudahkan mahasiswa untuk memahami materi perkuliahan dan dapat menerapkannya dalam proses perancangan. Sarana penunjang tersebut salah satunya adalah suatu perangkat praktikum yang terstruktur metode dan prosedurnya.

Sehubungan dengan kompleksnya masalah-masalah keandalan dalam proses perancangan dan banyaknya struktur bangunan laut, sehingga tidak memungkinkan untuk menjadikan semuanya sebagai bahan praktikum.

Untuk itu perlu dipilih beberapa studi kasus yang dianggap dapat mewakili.

Praktikum yang direncanakan (disusun), yaitu "PERANCANGAN DAN ANALISA KEANDALAN DENGAN BANTUAN KOMPUTER".

III.2. PENENTUAN STUDI KASUS

Dalam suatu proses perancangan sangat banyak kasus-kasus yang berkaitan dengan mata kuliah analisa keandalan dan resiko, dimana kasus-kasus tersebut perlu dipilih karena tidak memungkinkan untuk memasukkan semua kasus tersebut. Pemilihan kasus didasarkan pada masalah-masalah yang sering ditemui di lapangan dan yang berkaitan dengan struktur bangunan laut.

Kasus pertama menganalisa keandalan sebuah beam yang dikenai beban momen bending. Dengan melakukan normalisasi terhadap distribusi-distribusi masing-masing variabelnya akan didapatkan bidang kegagalan normalisasi pada sistem koordinat z . Pada studi kasus ini mahasiswa akan belajar menormalisasi distribusi variabel, baik itu variabel yang bersifat *corellated* maupun yang *uncorellated*. Dengan melakukan praktikum pada kasus ini secara berulang-ulang mahasiswa akan mendapatkan pengalaman cara menentukan harga awal yang paling baik untuk sebuah iterasi.

Untuk praktikum II, studi kasus diambilkan contoh masalah yang berkaitan dengan perencanaan bangunan laut. Bangunan laut yang dipilih disini adalah struktur jacket, dimana studi kasus yang dipilih adalah menghitung keandalan sambungan Y (*Y-Joint*). Pemilihan studi kasus ini berkenaan dengan sangat pentingnya keandalan sambungan pada struktur jacket, dimana dalam struktur suatu jacket 80% terdiri dari chord dan brace yang dihubungkan dengan sambungan.

Sedangkan untuk kasus terakhir merupakan studi kasus yang berkenaan dengan kasus *Fault Tree Analysis*. Seperti telah dijabarkan diatas, metode ini mencari penyebab sebuah kejadian yang tidak diinginkan, yang disebut sebagai *top event*. Pekerjaan selanjutnya dilakukan dengan arah kebawah, memecah sistem dalam detail yang terus bertambah untuk menentukan akar penyebab atau kombinasi dari penyebab-penyebab terjadinya *top event*. *Top event* biasanya adalah kegagalan yang membawa konsekuensi paling besar yang menimbulkan bahaya serius bagi keselamatan atau kerugian ekonomis yang penting.

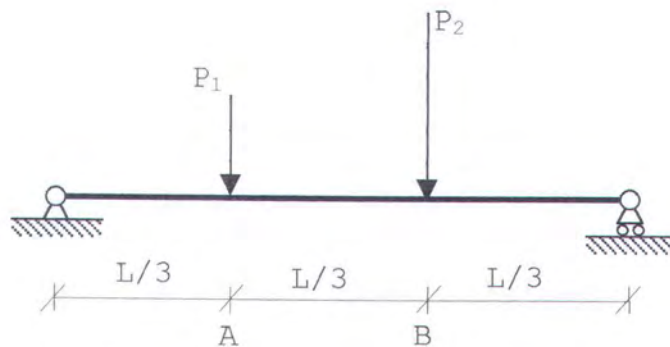
Kasus yang akan diambil disini adalah sistem listrik pada sebuah jacket. Dengan mengerjakan studi kasus ini diharapkan mahasiswa dapat mempelajari hubungan-hubungan setiap komponen dalam suatu sistem,

dan dapat menyusun suatu pohon kegagalan mulai dari *basic event* sampai *top event*. Selain itu mahasiswa juga diharapkan mempelajari komponen-komponen mana yang paling berpengaruh terhadap runtuhnya sistem tersebut.

III.2.1 Kasus I

Dari uraian diatas, kasus I [Thoft-Christensen & Murotsu (1986)] dapat didefinisikan sebagai berikut:

"MENENTUKAN INDEKS KEANDALAN BALOK AKIBAT MOMEN BENDING MAKSIMUM DENGAN MENGGUNAKAN AFOSM."



Gambar 3.1 Skema kasus I

Gambar diatas menunjukkan dua beban terpusat p_1 dan p_2 bekerja pada sebuah balok yang diasumsikan mengalami kegagalan pada $|m| \geq m_F$, dimana m_F adalah momen kritis dan m adalah momen maksimum balok.

Diketahui:

$$\mu P_1 = 10 \text{ kN}$$

$$\sigma P_1 = 0.283 \text{ kN}$$

$$\mu P_1 = 10 \text{ kN} \qquad \sigma P_1 = 0.283 \text{ kN}$$

$$\mu P_2 = 17 \text{ kN} \qquad \sigma P_2 = 0.500 \text{ kN}$$

$$\mu L = 9 \text{ m} \qquad \sigma L = 0.5 \text{ m}$$

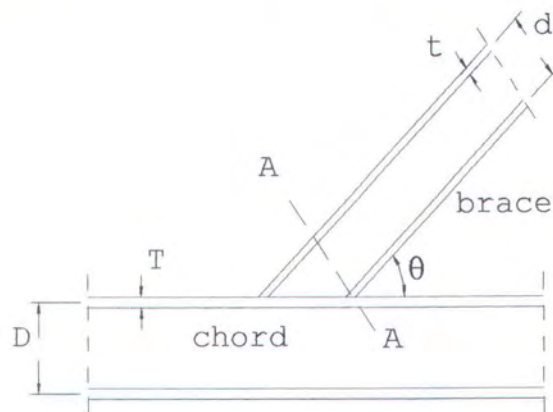
$$\mu M_F = 60 \text{ kNm} \qquad \sigma M_F = 6 \text{ kNm}$$

Semua variabel diasumsikan uncorellated kecuali P_1 dan P_2 , dengan $\text{Cov}[P_1, P_2] = 0.14 \text{ kN}^2$

III.2.2 Kasus II

Kasus II [Thoft-Christensen & Murotsu (1986)] dapat didefinisikan sebagai berikut:

"BAGAIMANA MENENTUKAN PELUANG KEANDALAN SAMBUNGAN PIPA (Y-JOINT) AKIBAT *IN-PLANE BENDING MOMENT* DAN *OUT-PLANE BENDING MOMENT* DENGAN MENGGUNAKAN SIMULASI MONTE CARLO"



Gambar 3.2 Skema kasus II

Dari gambar diatas efek beban yang bekerja pada penampang A-A adalah gaya aksial N , *in-plane bending moment* M_i dan *out-plane bending moment* M_o .

Batas keamanan (*safety margin*) dari sambungan diatas adalah:

$$M = 1 - 2 \cdot 10^{-5} \cdot N - (2 \cdot 10^{-4} |M_i|)^{1.2} - (2 \cdot 10^{-5} |M_o|)^{2.1}$$

Dimana, $\mu_N = 10^4 N$

$$\sigma_N = 0.2 \cdot 10^4 N$$

$$\mu_{M_i} = 2 \cdot 10^3 Nm$$

$$\sigma_{M_i} = 0.5 \cdot 10^3 Nm$$

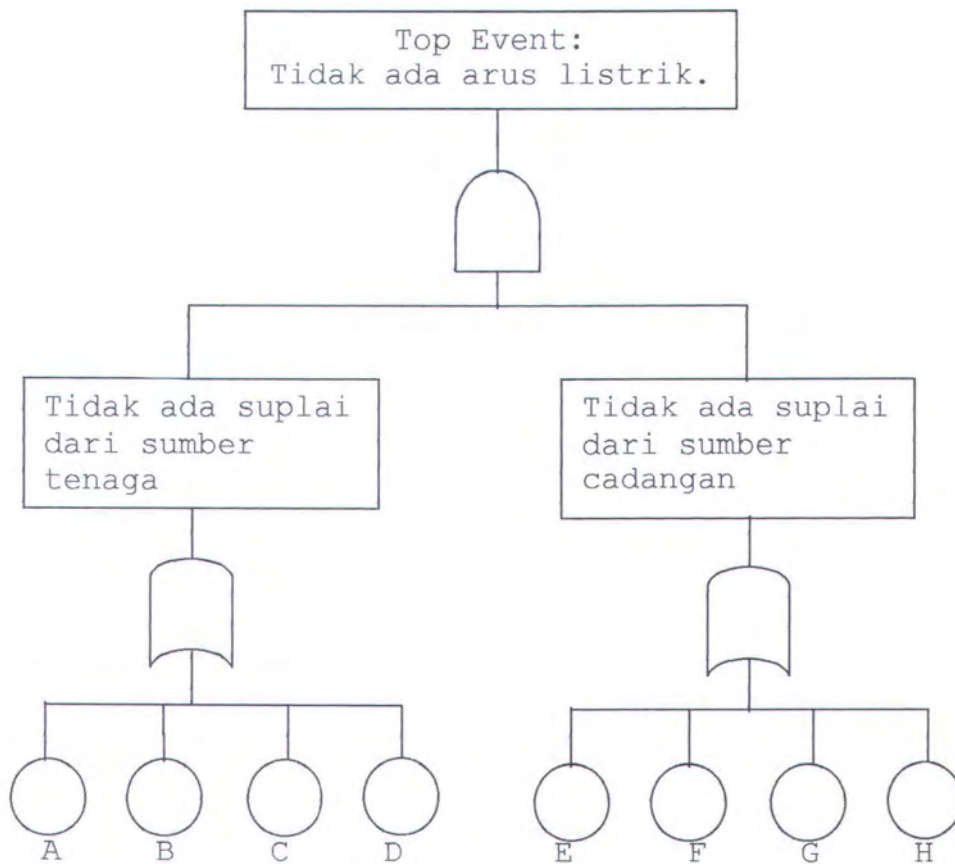
$$\mu_{M_o} = 10^4 Nm$$

$$\sigma_{M_o} = 0.1 \cdot 10^4 Nm$$

III.2.3 Kasus III

Kasus III [Ramakumar.(1993)] dapat didefinisikan sebagai berikut:

"BAGAIMANA MENENTUKAN/MENGHITUNG PELUANG TOP EVENT,
DARI SEBUAH SISTEM LISTRIK PADA JACKET"



Gambar 3.3 Diagram pohon kasus III

Keterangan:

- A: Kerusakan pada generator utama.
- B: Kerusakan pada *circuit breaker*.
- C: Kerusakan pada jaringan listrik.
- D: Kerusakan pada sekring.

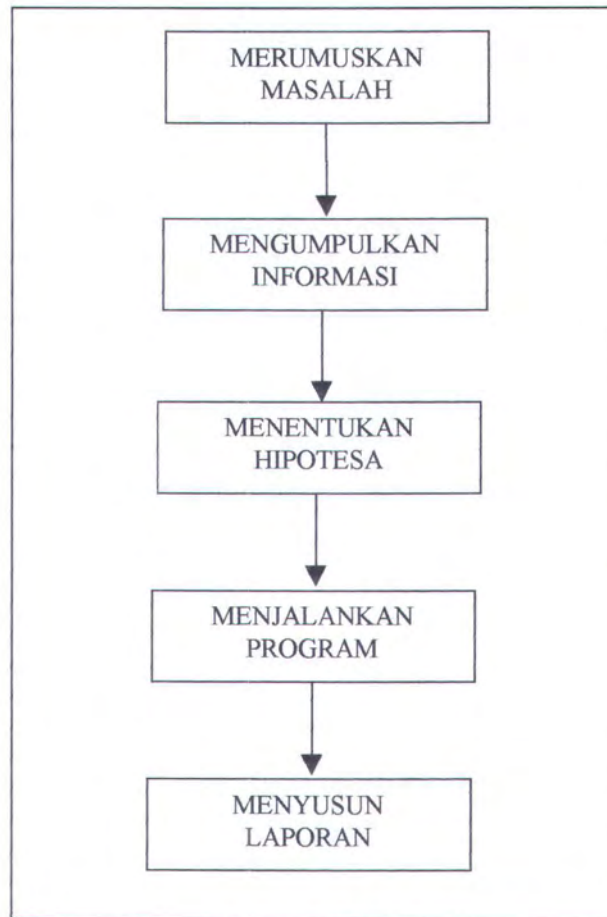
- E: Kerusakan pada generator cadangan.
- F: Kerusakan pada *circuit breaker*.
- G: Kerusakan pada jaringan listrik.
- H: Kerusakan pada sistem otomatisasi

III.3 PENENTUAN PROSEDUR PRAKTIKUM

Praktikum merupakan sarana yang baik bagi mahasiswa untuk melakukan latihan-latihan dalam rangka mengaplikasikan atau menerapkan teori-teori atau materi-materi perkuliahan dalam kasus-kasus nyata yang sering dijumpai di lapangan pekerjaan. Kelakuan coba-coba saja dalam praktikum tidaklah cukup, karenanya harus disusun suatu prosedur praktikum yang baik dan jelas. Penyusunan prosedur praktikum ini akan sangat membantu mahasiswa dalam memahami dan mewujudkan tujuan-tujuan dari setiap praktikum.

Prosedur praktikum disusun berdasarkan tahapan atau langkah-langkah praktikum, sehingga mahasiswa benar-benar bisa memahami esensi dari praktikum dalam tiap tahapnya. Mahasiswa bisa mengerti dan memahami pada langkah/tahap pertama apa saja yang harus dilakukan, kemudian pada tahap berikutnya apa yang harus dipersiapkan dan seterusnya.





Gambar 3.1 Prosedur Praktikum

III.3.1 Perumusan Masalah

Langkah pertama yang harus dilakukan dalam praktikum ini adalah merumuskan masalah, yaitu mengidentifikasi kasus-kasus yang telah ditentukan. Kasus-kasus yang telah ditentukan tersebut masih bersifat umum sehingga masih perlu diidentifikasi agar kasus bisa lebih jelas dan dapat diselesaikan dengan mudah. Pada langkah inilah mahasiswa belajar bagaimana merumuskan atau menyederhanakan masalah-masalah yang

bersifat umum (komplek) menjadi yang sederhana dan dalam bentuk baku sehingga mudah diselesaikan.

Agar perumusan masalah ini lebih terarah, mahasiswa harus mencari informasi dan mengumpulkan informasi-informasi/data-data penunjang dari berbagai literatur atau referensi yang sesuai. Pada langkah ini mahasiswa seharusnya sudah dapat menentukan masukan-masukan atau paling tidak sudah mempunyai gambaran masukan atau data apa saja yang dibutuhkan untuk menyelesaikan kasus.

III.3.2 Pengumpulan Informasi

Setelah masalah dirumuskan langkah selanjutnya adalah mengumpulkan informasi yang berkaitan dengan masalah tersebut dari beberapa referensi, baik itu mengenai teori-teori dasar penunjang maupun peraturan-peraturan yang sesuai yang telah ditetapkan. Pengumpulan informasi ini erat kaitannya dengan langkah berikutnya yaitu menentukan hipotesa. Semakin banyak informasi yang berhasil dikumpulkan maka akan semakin akurat pula hipotesa yang akan ditentukan nantinya. Informasi yang dikumpulkan harus akurat dan dapat dipertanggungjawabkan keabsahannya.

III.3.3 Penentuan Hipotesa

Dari informasi yang telah dikumpulkan, maka disusunlah hipotesa dari masalah/kasus tersebut. Hipotesa ini merupakan praduga awal yang masih perlu dibuktikan kebenarannya. Hipotesa ini biasanya ditentukan selain berdasarkan referensi tertulis yang ada juga berdasarkan pada pengalaman-pengalaman.

Dalam penentuan hipotesa ini dituntut kemampuan dalam memahami informasi-informasi yang diperoleh dan menghubungkannya dengan masalah yang ada. Disini peran pengalaman cukup besar, seseorang yang telah berpengalaman akan lebih mudah menentukan hipotesa dan biasanya lebih akurat.

III.3.4 Pelaksanaan Praktikum

Sebelum praktikum dilaksanakan mahasiswa harus menyusun atau merencanakan terlebih dahulu langkah-langkah yang akan dilaksanakan selama praktikum. Hal ini dimaksudkan agar praktikum dapat dilaksanakan secara sistematis tahap demi tahap.

Praktikum dilaksanakan dengan memasukkan data-data yang telah diperoleh (dibutuhkan) pada program yang telah disiapkan. Praktikum ini untuk menunjukkan atau membuktikan apakah hipotesa yang telah ditentukan sebelumnya itu benar atau tidak. Praktikum bisa

dilaksanakan berkali-kali dengan variasi data untuk mendapatkan hasil yang memuaskan serta memperdalam pemahaman.

III.3.5 Penyusunan Laporan

Setiap kali selesai pelaksanaan praktikum, haruslah disusun laporan yang berisi antara lain: teori-teori dasar, perumusan masalah, analisa, evaluasi, kesimpulan dan saran. Dari laporan ini dapat dilihat sejauh mana seorang mahasiswa dapat memahami dan menguasai jalannya praktikum serta tujuan praktikum yang telah dilaksanakan.

Analisa yang dilakukan dapat dijadikan sebagai bahan evaluasi. Sejauh mana seorang praktikan dapat memahami teori-teori yang telah diterimanya selama mengikuti perkuliahan, dan bagaimana praktikan menerepkan teori-teori tersebut dalam menyelesaikan masalah yang ada.

III.4 PENENTUAN PENYELESAIAN STUDI KASUS

Untuk menyelesaikan studi kasus yang dipilih (ditentukan) maka disusun program dengan menggunakan Borland Delphi. Dalam penyusunan program ini harus memperhatikan aturan-aturan yang sesuai (telah ditentukan). Untuk kasus I mengacu pada bentuk baku

dari metode *Advance First Order Second Moment* dan untuk kasus II adalah berdasarkan bentuk baku dari metode Simulasi Monte Carlo sedangkan kasus III berdasarkan pada metode *Fault Tree Analysis*.

Adapun urutan-urutan (langkah-langkah) penyelesaian kasus adalah sebagai berikut.

III.4.1 Penyelesaian Studi Kasus I

Untuk menyelesaikan studi kasus ini dapat diikuti langkah-langkah sebagai berikut:

Dari keadaan pembebanan pada kasus I, didapatkan momen maksimum terletak pada titik B sebesar:

$$m = \frac{1}{9} L(p_1 + 2p_2) = \frac{1}{9} Lp,$$

Bidang kegagalannya:

$$f'(m_F, L, p) = m_F - \frac{1}{9} Lp = 0$$

p menyatakan *random variable* $P=P_1+2P_2$, bila p_1 dan p_2 *corellated* dengan $cov[p_1, p_2]=0.14 \text{ kN}^2$, maka:

$$\mu_P = \mu_{p_1} + 2 \cdot \mu_{p_2}$$

$$\mu_P = 10 + 2 \cdot 17 = 44 \text{ kN}, \text{ dan}$$

$$\sigma_P^2 = \sigma_{p_1}^2 + 2^2 \cdot \sigma_{p_2}^2 + 2^2 \cdot COV[p_1, p_2]$$

$$\sigma_p = (0,08 + 4 \cdot 0.25 + 4 \cdot 0.14)^{\frac{1}{2}} = 1.28 \text{ kN}$$

Dari bidang kegagalan tersebut dilakukan normalisasi dengan pers. 2.46,

$$Z_1 = \frac{M_F - \mu M_F}{\sigma M_F}$$

$$Z_2 = \frac{L - \mu L}{\sigma L}$$

$$Z_3 = \frac{p - \mu p}{\sigma p}$$

$$M_F = 6 \cdot Z_1 + 60$$

$$L = 0.5 \cdot Z_2 + 9$$

$$p = 1.28 \cdot Z_3 + 44$$



dari transformasi diatas akan didapatkan normalisasi safety margin,

$$f : Z_1 \cdot \sigma_{M_F} + \mu_{M_F} - \frac{1}{9} (Z_2 \cdot \sigma_L + \mu_L) \cdot (Z_3 \cdot \sigma_p + \mu_p) = 0$$

$$f : (60 + 6Z_1) - \frac{1}{9} (9 + 0.8Z_2) \cdot (44 + 1.28Z_3) = 0$$

$$f : 144 + 54\beta\alpha_1 - 22\beta\alpha_2 - 11.52\beta\alpha_3 - 0.64\beta^2\alpha_2\alpha_3 = 0$$

dengan substitusi dengan pers. 2.51 didapatkan hasil

$$\beta = \frac{144}{-54\alpha_1 + 22\alpha_2 + 11.52\alpha_3 + 0.64\beta\alpha_2\alpha_3}$$

dimana, $\alpha_i = -\frac{1}{k} \cdot \frac{\partial f}{\partial Z_i} (\beta\alpha^-)$ $i=1, \dots, n$

$$\alpha_1 = -\frac{54}{k},$$

$$\alpha_2 = \frac{1}{k} (22 + 0.64\beta\alpha_3),$$

$$\alpha_3 = \frac{1}{k} (11.52 + 0.64\beta\alpha_2)$$

$$\text{dan, } k = \left[\sum_{j=1}^n \left(\frac{\partial f}{\partial Z_j} (\beta\alpha) \right)^2 \right]^{\frac{1}{2}}$$

$$k = \left[\left(\frac{\partial f}{\partial Z_1} \right)^2 + \left(\frac{\partial f}{\partial Z_2} \right)^2 + \left(\frac{\partial f}{\partial Z_3} \right)^2 \right]^{\frac{1}{2}}$$

$$k = \left[(54)^2 + (22 + 0.64\beta\alpha_3)^2 + (11.52 + 0.64\beta\alpha_2)^2 \right]^{\frac{1}{2}}$$

selanjutnya dengan iterasi akan didapatkan harga β .

III.4.2 Penyelesaian studi kasus II.

Simulasi Monte Carlo adalah metode keandalan yang melakukan perhitungan terhadap fungsi kegagalan f dengan menggunakan *random number generator*. Artinya untuk sebuah fungsi kegagalan f , dimana untuk setiap realisasi perubah acak dalam fungsi tersebut disimulasikan dengan bantuan RNG dari sebuah komputer. Penyelesaian studi kasus II dengan menggunakan simulasi monte carlo dapat dilakukan sesuai tahapan-tahapan berikut ini.

Sebuah fungsi kegagalan,

$$M = 1 - 2 \cdot 10^{-5} \cdot N - (2 \cdot 10^{-4} |Mi|)^{1.2} - (2 \cdot 10^{-5} |Mo|)^{2.1}$$

Sejumlah angka acak dengan fkp uniform yang dikeluarkan RNG dinyatakan A_p , $p=1, \dots, m$. Selanjutnya angka acak A_p ditransformasikan menjadi X_p , $p=1, \dots, m$.

Transformasi A_p menjadi X_p dilakukan secara numerik dengan prosedur sebagai berikut:

Untuk setiap fkp dari X_p yang telah diketahui distribusinya dilakukan pembagian rentang dalam I interval yang sama lebar sepanjang dx .

Setiap daerah pembagian tersebut dihitung luasannya yang merupakan peluang X_p dalam interval tersebut sebesar P_i .

Apabila suatu A_p berada dalam batas interval yang sesuai $P_i < A_p < P_{i+1}$ maka A_p dipahami telah ditransformasikan sebagai X_i .

Hasil dari transformasi tersebut diatas dimasukkan ke dalam fungsi kegagalan yang telah ditentukan, untuk setiap hasil yang negatif dicatat sebagai kegagalan. Pencatatan hasil kegagalan yang terjadi dilakukan terhadap semua X_i .

Selanjutnya peluang kegagalan dari fungsi tersebut adalah perbandingan antara jumlah percobaan dengan jumlah kegagalan yang dihasilkan.

III.4.3 Penyelesaian studi kasus III

Penyelesaian studi kasus III dengan menggunakan *Fault Tree Analysis* adalah sebagai berikut:

1. Pertama kali disusun sebuah pohon kegagalan sesuai dengan susunan sistem yang telah ditentukan.
2. Untuk setiap *Basic Event* harus sudah diketahui nilai peluang kegagalannya.
3. Dengan memperhatikan nilai peluang yang telah diketahui dan hubungan pada kelompok kejadian yang sedang ditinjau dapat ditentukan nilai peluang kegagalan pada kejadian berikutnya.
4. Langkah No. 3 dilakukan berturut-turut terhadap setiap kelompok kejadian sehingga didapatkan nilai peluang kegagalan untuk *Top Event*-nya.



BAB IV PRAKTIKUM

Dan Allah menurunkan dari langit air (hujan) dan dengan air itu dihidupkan-Nya bumi sesudah matinya. Sesungguhnya pada yang demikian itu terdapat tanda-tanda (kebesaran Tuhan) bagi orang-orang yang mendengarkan (pelajaran) (QS. An Nahl: 65)

BAB IV

PRAKTIKUM

Sebagai penunjang mata kuliah analisa keandalan dan resiko disusun praktikum dengan tiga metode, yaitu metode *Advanced First Order Second Moment* dan Simulasi Monte Carlo untuk menyelesaikan masalah/kasus tunggal sedangkan *Fault Tree Analysis* untuk menyelesaikan kasus jamak.

Untuk praktikum I dan II disusun sebuah program yang dapat digunakan untuk menyelesaikan kasus-kasus tunggal, metode AFOSM digunakan untuk menyelesaikan kasus yang fungsi kinerjanya linier atau sedikit non-linier sedangkan Simulasi Monte Carlo digunakan untuk menyelesaikan kasus yang fungsi kinerjanya sangat non-linier.

Untuk praktikum III program yang disusun adalah untuk menyelesaikan sebuah kasus sebuah sistem yang terdiri dari beberapa komponen yang sangat mempengaruhi komponen utama.

Secara umum tujuan yang ingin dicapai dalam ketiga praktikum ini adalah sebagai berikut:

1. Melatih mahasiswa untuk lebih memahami dan mengerti tentang teori-teori dan kaidah-kaidah analisa keandalan dan resiko.
2. Melatih mahasiswa untuk belajar mengintegrasikan beberapa teori analisa keandalan dan resiko untuk menerapkannya pada masalah nyata.
3. Melatih mahasiswa untuk merumuskan permasalahan-permasalahan analisa keandalan dan resiko yang sering ditemui dilapangan.
4. Merangsang mahasiswa untuk lebih bersemangat dalam mempelajari dan menekuni mata kuliah analisa keandalan dan resiko.
5. Merangsang mahasiswa untuk lebih memanfaatkan segala fasilitas yang ada di kampus.

IV.1 PRAKTIKUM I

Pada praktikum I ini yang akan dilakukan yaitu segala masalah tunggal yang fungsi kinerjanya linier atau sedikit non-linier.

IV.1.1 Tujuan praktikum I

1. Mahasiswa mengerti dan memahami prosedur perhitungan keandalan dengan menggunakan metode AFOSM.

2. Mahasiswa mengerti dan memahami data yang diperlukan dalam perhitungan keandalan dengan menggunakan metode AFOSM.
3. Mahasiswa dapat memahami dan mengerti alasan mengapa dipilih kasus ini sebagai praktikum I.
4. Mahasiswa dapat melakukan perhitungan keandalan dengan menggunakan metode AFOSM.

IV.1.2 Petunjuk Pelaksanaan

Sebelum praktikum dilaksanakan, ada beberapa tugas pendahuluan yang harus diselesaikan yaitu:

a. Merumuskan masalah

Tugas ini adalah menentukan dan mengidentifikasi masalah yang akan dijadikan praktikum yang sesuai dengan batasan-batasan yang telah ditentukan.

b. Mengumpulkan informasi

Dua referensi utama yang harus dipelajari (dibaca) yaitu: "*Application of Structural System Reliability Theory*", [Thoft-Christensen & Murotsu (1986)] dan "*Pengantar Rekayasa Keandalan*", [Rosyid (1995)]. Karena praktikum yang disusun ini berdasarkan pada contoh dan perumusan baku yang berada dalam referensi tersebut dan

jika dirasa perlu bisa ditambah dengan referensi-referensi yang lain yang sesuai dengan masalah yang ditentukan. Dari informasi-informasi yang dikumpulkan, mahasiswa dapat memahami alasan setiap pengambilan keputusan (pilihan) selama pelaksanaan praktikum.

c. Menentukan hipotesa

Dari masalah yang telah dirumuskan dan informasi yang telah dikumpulkan, dicoba untuk menentukan sebuah hipotesa. Hipotesa ini bisa berupa harga awal untuk variabel-variabel masukan.

Setelah ketiga tugas pendahuluan ini diselesaikan barulah praktikum I dapat dilaksanakan.

d. Masukan data

Untuk memasukkan data/input pada tampilan terdapat beberapa pilihan, yaitu Praktikum I, Hasil praktikum I, Cetak Hasil Praktikum I.

1. Praktikum I

Untuk memasukkan data input dan memulai perhitungan keandalan. Dimana data yang dibutuhkan adalah harga awal untuk betha dan alpha.

2. Hasil Praktikum I

Untuk mengetahui hasil program yang telah dijalankan pada setiap iterasi.

3. Cetak Hasil Praktikum I

Untuk selanjutnya dilakukan proses pencetakan hasil dari program yang dilaksanakan untuk melengkapi laporan yang akan disusun setelah melakukan praktikum.

IV.2 PRAKTIKUM II

Pada praktikum II ini yang akan dilakukan yaitu segala masalah tunggal yang fungsi kinerjanya sangat non-linier.

IV.2.1 Tujuan praktikum II

1. Mahasiswa mengerti dan memahami prosedur perhitungan keandalan dengan menggunakan Simulasi Monte Carlo.
2. Mahasiswa mengerti dan memahami data yang diperlukan dalam perhitungan keandalan dengan menggunakan Simulasi Monte Carlo.
3. Mahasiswa dapat memahami dan mengerti alasan mengapa dipilih kasus ini sebagai praktikum II.
4. Mahasiswa dapat melakukan perhitungan keandalan dengan menggunakan Simulasi Monte Carlo.

IV.2.2 Petunjuk Pelaksanaan

Sebelum praktikum dilaksanakan, ada beberapa tugas pendahuluan yang harus diselesaikan yaitu:

a. Merumuskan masalah

Tugas ini adalah menentukan dan mengidentifikasi masalah yang akan dijadikan praktikum yang sesuai dengan batasan-batasan yang telah ditentukan.

b. Mengumpulkan informasi

Referensi utama yang harus dipelajari (dibaca) yaitu: "*Application of Structural System Reliability Theory*", [Thoft-Christensen & Murotsu (1986)] dan "Pengantar Rekayasa Keandalan", [Rosyid (1995)] serta "*An Introduction to Structural Reliability Theory*", [Ship Structure Committee (1990)]

Karena praktikum yang disusun ini berdasarkan pada contoh dan perumusan baku yang berada dalam referensi tersebut dan jika dirasa perlu bisa ditambah dengan referensi-referensi yang lain yang sesuai dengan masalah yang yang ditentukan. Dari informasi-informasi yang dikumpulkan, mahasiswa dapat memahami alasan setiap pengambilan keputusan (pilihan) selama pelaksanaan praktikum.

c. Menentukan hipotesa

Dari masalah yang telah dirumuskan dan informasi yang telah dikumpulkan, dicoba untuk menentukan sebuah

hipotesa. Hipotesa ini bisa berupa peluang keandalan untuk setiap jumlah percobaan yang diinginkan.

Setelah ketiga tugas pendahuluan ini diselesaikan barulah praktikum II dapat dilaksanakan.

d. Masukan data

Untuk memasukkan data/input pada tampilan terdapat beberapa pilihan, yaitu Praktikum II, Hasil praktikum II, Cetak Hasil Praktikum II.

1. Praktikum II

Untuk memasukkan data input dan memulai perhitungan keandalan. Dimana data yang dibutuhkan adalah:

- Jumlah Percobaan
- Jumlah variabel acak dasar
- Nilai rata-rata, simpangan baku dan distribusi untuk setiap variabel acak baku.

2. Hasil Praktikum II

Untuk mengetahui hasil program yang telah dijalankan pada setiap perhitungan.

3. Cetak Hasil Praktikum II

Selanjutnya dilakukan proses pencetakan hasil dari program yang dilaksanakan untuk melengkapi laporan yang akan disusun setelah melakukan praktikum.

IV.3 PRAKTIKUM III

Pada praktikum III ini yang akan dilakukan yaitu segala masalah jamak untuk sebuah sistem yang terdiri dari beberapa komponen.

IV.3.1 Tujuan praktikum III

1. Mahasiswa mengerti dan memahami prosedur perhitungan keandalan dengan menggunakan *Fault Tree Analysis*.
2. Mahasiswa mengerti dan memahami data yang diperlukan dalam perhitungan keandalan dengan *Fault Tree Analysis*.
3. Mahasiswa dapat memahami dan mengerti alasan mengapa dipilih kasus ini sebagai praktikum III.
4. Mahasiswa dapat melakukan perhitungan keandalan dengan menggunakan *Fault Tree Analysis*.

IV.3.2 Petunjuk Pelaksanaan

Sebelum praktikum dilaksanakan, ada beberapa tugas pendahuluan yang harus diselesaikan yaitu:

a. Merumuskan masalah

Tugas ini adalah menentukan dan mengidentifikasi masalah yang akan dijadikan praktikum yang sesuai dengan batasan-batasan yang telah ditentukan.

b. Mengumpulkan informasi

Referensi utama yang harus dipelajari (dibaca) yaitu: "*Engineering Reliability: Fundamentals and Applications*", [Ramakumar (1993)] dan "*Pengantar Rekayasa Keandalan*", [Rosyid (1995)] serta "*Practical Reliability Engineering*", [O'connor (1991)].

Karena praktikum yang disusun ini berdasarkan pada contoh dan perumusan baku yang berada dalam referensi tersebut dan jika dirasa perlu bisa ditambah dengan referensi-referensi yang lain yang sesuai dengan masalah yang ditentukan. Dari informasi-informasi yang dikumpulkan, mahasiswa dapat memahami alasan setiap pengambilan keputusan (pilihan) selama pelaksanaan praktikum.

c. Menentukan hipotesa

Dari masalah yang telah dirumuskan dan informasi yang telah dikumpulkan, dicoba untuk menentukan sebuah hipotesa. Hipotesa ini bisa berupa penentuan komponen penyebab kegagalan pada *top event*.

Setelah ketiga tugas pendahuluan ini diselesaikan barulah praktikum III dapat dilaksanakan.

d. Masukan data

Untuk memasukkan data/input pada tampilan terdapat beberapa pilihan, yaitu Praktikum III, Hasil praktikum III, Cetak Hasil Praktikum III.

1. Praktikum III

Untuk memasukkan data input dan memulai perhitungan keandalan. Dimana data yang dibutuhkan adalah:

- Nama kejadian.
- Jumlah anak cabang untuk setiap kejadian.
- Hubungan anak cabang untuk setiap kejadian.
- Peluang kegagalan bila kejadian adalah *basic event* (tidak mempunyai anak cabang).

2. Hasil Praktikum III

Untuk mengetahui hasil program yang telah dijalankan.

3. Cetak Hasil Praktikum III

Untuk selanjutnya dilakukan proses pencetakan hasil dari program yang dilaksanakan untuk melengkapi laporan yang akan disusun setelah melakukan praktikum III.

IV.4 PENULISAN LAPORAN

Untuk mengetahui kesungguhan dan pemahaman mahasiswa dalam melaksanakan praktikum maka setiap selesai praktikum diwajibkan untuk menyusun sebuah

laporan. Sistematika laporan disusun sesuai dengan prosedur praktikum yaitu pendahuluan, langkah-langkah percobaan, analisa dan penutup (kesimpulan dan saran).

a. Pendahuluan

Berisikan tentang latar belakang dan alasan pemilihan kasus, tujuan praktikum, teori dasar yang berhubungan dengan kasus yang dipilih, rumusan masalah dan identifikasi data serta ketentuan dasar atau persyaratan keandalan yang harus dipenuhi.

b. Langkah-langkah praktikum

Berisi langkah-langkah yang harus dilakukan selama praktikum, sesuai dengan prosedur praktikum yang telah ditentukan.

c. Hasil percobaan

Merupakan output dari program yang telah dijalankan, hasil percobaan ini bisa diperoleh dari beberapa data yang berbeda, semakin bervariasi data input semakin baik hasil yang akan diperoleh.

d. Analisa

Dari beberapa hasil percobaan dapat dilakukan analisa untuk mengetahui pengaruh dari tiap-tiap data masukan pada hasil percobaan, sehingga dapat diketahui

data mana yang paling menentukan dalam suatu proses perhitungan keandalan.

e. Penutup (kesimpulan dan saran)

Setelah dilakukan praktikum dengan beberapa variasi data maka praktikan harus dapat menarik suatu kesimpulan tentang praktikum tersebut. Kesimpulan tersebut hendaknya sesuai dengan tujuan yang ingin dicapai dalam pelaksanaan praktikum tersebut.





BAB V

KESIMPULAN DAN SARAN

Dan (ingatlah) ketika Luqman berkata kepada anaknya, diwaktu ia memberi pelajaran kepadanya: “Hai anakku, janganlah kamu mempersekutukan Allah, sesungguhnya mempersekutukan (Allah) adalah nyata-nyata kezaliman besar. (QS. Luqman: 13)

BAB V

KESIMPULAN DAN SARAN

Penulisan tugas akhir ini lebih ditujukan pada penyediaan perangkat praktikum untuk mata kuliah Analisa Keandalan dan Resiko, sehingga analisa-analisa yang dilakukan tidak terlalu detail. Dari penulisan tugas akhir ini dapat ditarik kesimpulan sebagai berikut:

1. Pemilihan jenis praktikum disamping mengandung beberapa substansi mata kuliah juga harus dipertimbangkan kondisi-kondisi lapangan, sehingga mampu memberikan gambaran yang lebih nyata terhadap praktikan.
2. Pemilihan studi kasus sebagai bahan praktikum harus bisa memberikan gambaran bagi mahasiswa tentang kasus-kasus yang sering muncul dilapangan.
3. Tujuan dari praktikum harus sejalan dengan tujuan mata kuliah yang bersangkutan.
4. Untuk praktikum yang melakukan proses iterasi perlu dilakukan pemilihan harga awal secara teliti untuk mendapatkan hasil yang akurat.

Dengan melihat tujuan yang dapat dicapai dari pelaksanaan praktikum, seperti telah diuraikan pada bab sebelumnya, penulis ingin memberikan saran sebagai berikut:

1. Untuk mendapatkan hasil yang baik praktikan seharusnya melaksanakan praktikum sesuai prosedur yang telah ditentukan dan melakukannya berulang-ulang dengan studi kasus yang berbeda-beda pula.
2. Untuk lebih memaksimalkan pemahaman tentang mata kuliah yang bersangkutan perlu kiranya mengembangkan praktikum ini dengan memperluas kasus.
3. Akan lebih baik bila semua mata kuliah di Teknik Kelautan ditunjang dengan perangkat praktikum.
4. Keterkaitan antara satu mata kuliah dengan yang lain akan lebih mudah dimengerti melalui praktikum, bila mengambil kasus yang lebih nyata dilapangan.



DAFTAR PUSTAKA

Dan diantara tanda-tanda kekuasaan-Nya ialah bahwa Dia mengirimkan angin sebagai pembawa berita gembira dan untuk merasakan kepadamu sebagian dari rahmat-Nya, dan supaya kapal-kapal dapat berlayar dengan perintah-Nya dan lagi agar kamu mencari sebagian dari karunia-karunia-Nya (dengan berniaga melalui laut), dan supaya kamu bersyukur (lalu beriman) (QS. Ar Rum: 46)

DAFTAR PUSTAKA

- AME, Reliability Analysis Applications to Structures,
Advanced Mechanics & Engineering Ltd.
Guildford.
- Amien M., (1988), Buku Pedoman Laboratorium dan
Petunjuk Praktikum Pendidikan IPA Umum (General
Science) untuk Lembaga Pendidikan Tenaga
Kependidikan, Dep. Pendidikan dan Kebudayaan
Dirjen. Pendidikan Tinggi, Proyek Pengembangan
Lembaga Pendidikan Tenaga Kependidikan,
Jakarta.
- Jogiyanto H. M., (1993), Turbo Pascal, Andi Offset,
Yogyakarta.
- Matcho J. dan Faulkner D. R., (1997), Special Edition
Using Delphi, Simon & Schuster (Asia) Pte. Ltd
- O'connor P.D.T., (1991), Practical Reliability
Engineering, John Willey & Sons Ltd.,
Chicester.
- Pranata A., (1997), Pemrograman Borland Delphi, Andi
Offset, Yogyakarta.
- Ramakumar R., (1993), Engineering Reliability:
Fundamentals and Applications, Prentice-Hall
International, Inc.

- Rooijackers, Ad., (1990), Mengajar Dengan Sukses, Gramedia, Jakarta.
- Rosyid D.M., (1995), Pengantar Rekayasa Keandalan, FTK-ITS, Surabaya.
- Santoso P.I., (1993), Dasar-dasar Pemrograman Pascal Teori dan Program Terapan, Andy Offset, Yogyakarta.
- Santoso P.I., (1995), Struktur Data Menggunakan Turbo Pascal 6.0, Andi Offset, Yogyakarta.
- Ship Structure Committee, (1990), An Introduction to Structural Reliability Theory, SSC-351.
- Thoft-Christensen, Palle dan Murotsu, Yoshisada, (1986), Application of Structural System Reliability Theory, Springer-Verlag Berlin, Heidelberg.
- Todd B. and Kellen V., (1995), DELPHI a Developer's Guide, M&T Books, New York.
- Utomo T. dan Ruijter K., (1991), Peningkatan dan Pengembangan Pendidikan, Gramedia Pustaka Utama, Jakarta.
- Vrouwenvelder, A., Karadeniz, H. dan Van Manen, S., Reliability Analysis for the Fatigue Limit State of Offshore Structures, Institute TNO for Building Materials and Building Structures, no.B-84-469.

Walpole R.E., Myers R.H., (1986), Ilmu Peluang dan Statistik untuk Insinyur dan Ilmuwan, Penerbit ITB, Bandung.

Watson H.J., Blackstone J.H. Jr., (1981), Computer Simulation, John Willey & Sons.

Wozniewicz A. and Shammass N., (1995), Teach Yourself Borland Delphi in 21 Days, Sams Publishing, Indianapolis, USA



LAMPIRAN A PETUNJUK PEMAKAIAN

**(Tuhan) Yang Maha Pemurah. Yang telah mengajarkan Al-Qur'an.
Menciptakan manusia. Mengajarkan pandai bicara. (QS. Ar Rahman: 1-4)**



FAKULTAS TEKNOLOGI KELAUTAN ITS
PROGRAM STUDI TEKNIK LAUTAN

TUGAS - AKHIR.

No. : 131 /PT12.H8.FTK.4/Q/95

MATA KULIAH : TUGAS AKHIR
NOMOR MATA KULIAH : OE. 1701.....
NAMA MAHASISWA : Edi Djoko Firjanto.....
NOMOR POKOK : 490.4300.136.....
TANGGAL DIBERIKAN TUGAS : 13 Juli 1995.....
TANGGAL SELESAI TUGAS : 13 Desember 1995.....
DOSEN PEMBIMBING : DR. Ir. Daniel M. Rosyid.....
DR. Ir. E. B. Djatmiko, MSc.....

TEMA/URAIAN/DATA-DATA YANG DIBERIKAN :

JUDUL : " PENGEMBANGAN PERANGKAT LUNAK PENUNJANG
PRATIKUM MATA KULIAH ANALISA KEANDALAN
DAN RESIKO (OE. 1520) "

SURABAYA, 13 Juli 1995.....

DIBUAT RANGKAP 4 :

1. Mahasiswa ybs.
2. Dekan
3. Dosen Pembimbing.
4. Arsip Kajur.



FAKULTAS TEKNOLOGI KELAUTAN ITS
TEKNIK LAUTAN

Ir. E. B. Djatmiko, MSc.

NIP. 81 407 592

**JURUSAN TEKNIK KELAUTAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

LEMBAR PRESENSI KONSULTASI TUGAS AKHIR

NAMA MAHASISWA : Edi Djoko Firjanto
 NOMOR POKOK : 4390.100.008
 NAMA DOSEN PEMBIMBING : Ir. Daniel Moh. Rosyid Ph.D
 Ir. Eko Budi Djatmiko MSc., Ph.D
 TUGAS DIMULAI :
 DISELESAIKAN :
 JUDUL TUGAS AKHIR : Pengembangan Perangkat Lunak Penunjang
 Praktikum Mata Kuliah Analisa Keandalan
 dan Resiko.

NO	TANGGAL	KONSULTASI MENGENAI	TANDA TANGAN DOSEN PEMBIMBING
1	15/11/95	Rumusan Masalah	<i>[Signature]</i>
2	7/12/95	Revisi Rumusan Masalah	<i>[Signature]</i>
3	21/2/96	Studi Kasus	<i>[Signature]</i>
4	6/3/96	- Software requirement	<i>[Signature]</i>
4	19/3/96	- Algoritma dan programming	<i>[Signature]</i>
5	12/3/97	- Validasi Program	<i>[Signature]</i>
6	17/4/97	- Pembuatan TA	<i>[Signature]</i>
7	15/4/98	- Kumpulan	<i>[Signature]</i>
8	5/8/98	TA kelulusan	<i>[Signature]</i>

Catatan: Lembar presensi ini harus ditunjukkan pada saat ujian Tugas Akhir



DEPARTEMEN PENDIDIKAN DAN KEBUDAYAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI KELAUTAN
JURUSAN TEKNIK KELAUTAN

Kampus ITS - Sukolilo, Surabaya 60111 Telp. 5947254, 597274 psw. 144 Telex 34224 Fax 5947254

FORMULIR EVALUASI KEMAJUAN TUGAS AKHIR

Kami, dosen pembimbing Tugas Akhir dari mahasiswa :

Nama : EDI DJOKO FIRJANTO
NRP : 4300.100.008
Judul TA : PENGEMBANGAN PERANGKAT LUNAK PENUNJANG
PRAKTIKUM MATA KULIAH ANALISA KEANDALAN
DAN RESIKO.

Setelah mempertimbangkan butir-butir berikut :

- Keaktifan mahasiswa dalam mengadakan asistensi.
- Proporsi Materi TA yang telah diselesaikan sampai saat ini.
- Prospek penyelesaian TA dalam jangka waktu yang relevan.
- Masa Studi yang tersisa.

Dengan ini kami mengusulkan agar TA mahasiswa tersebut diputuskan untuk :

- ☐ Dibatalkan keseluruhannya dan mengajukan judul baru
- ☐ Diperkenankan menyelesaikan tanpa perubahan
- ☒ Diperkenankan mengikuti Ujian Tugas Akhir dengan Judul :
Pengembangan Perangkat Lunak Penunjang
Praktikum MK. Analisa Keandalan & Resiko

Selanjutnya mahasiswa diatas diharuskan untuk dapat menyelesaikan Tugas Akhirnya dan dapat mengikuti ujian Tugas Akhir untuk Wisuda September 1998

Surabaya, 6/8/ 1998
Dosen Pembimbing ;

(Signature)
(D.M. Rogius)

PETUNJUK PEMAKAIAAN

Program ini ditulis/disusun sebagai penunjang praktikum untuk mata kuliah analisa keandalan dan resiko dengan menggunakan bahasa pemrograman Borland Delphi 3, yang berbasiskan pada bahasa program Borland Pascal. Borland Delphi 3 adalah bahasa program pada sistem Windows yang berbasiskan 32 bit, artinya program ini hanya dapat berjalan pada komputer dengan sistem Windows NT atau Window 95 keatas. Penulisan program dilakukan dengan menggunakan komputer yang memiliki perangkat antar muka (VGA Card) untuk monitor dengan resolusi 800x600 dpi. Sehingga untuk mendapatkan hasil tampilan yang maksimal praktikum sebaiknya dilakukan pada resolusi 800x600 dpi atau lebih.

Setiap praktikan harus sudah mempersiapkan diri sebelum melaksanakan praktikum mulai dari pemahaman materi, mengerti prosedur praktikum, penguasaan masalah dan persiapan-persiapan yang lain seperti yang telah ditentukan sesuai dengan prosedur praktikum.

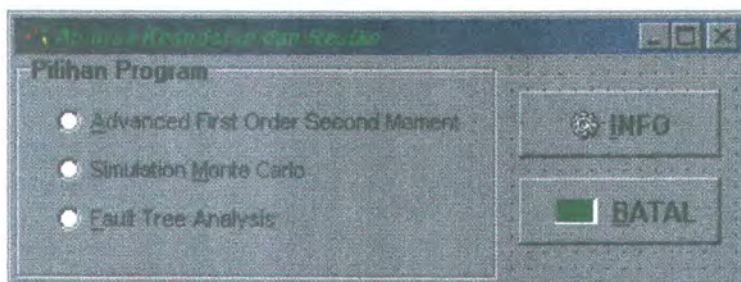
Praktikum harus dilaksanakan tahap per tahap sesuai dengan prosedur yang diberikan untuk setiap metode yang diambil. Setelah melakukan praktikum, praktikan harus dapat menyusun laporan praktikum untuk mengetahui sejauh mana praktikan mengerti/memahami kaidah-kaidah analisa keandalan dan sejauh mana dapat menerapkannya pada masalah-masalah nyata.

Praktikum terdiri atas tiga metode analisa keandalan yang paling sering dipakai, yaitu:

- 1 Advance First Order Second Moment.
- 2 Simulasi Monte Carlo.
- 3 Fault Tree Analysis.

Untuk setiap metode diberikan studi kasus yang berkaitan dengan perancangan pada sebuah bangunan lepas pantai.

Tampilan pertama memperlihatkan form pilihan tentang praktikum yang akan dilaksanakan.



Gambar tampilan program utama.



Untuk memilih praktikum cukup dengan meng-click salah satu tombol *radio button*, selanjutnya akan terlihat tampilan berikutnya yaitu form program sesuai pilihan.

1. Praktikum I

Pada praktikum I yang menggunakan metode pembahasan AFOSM, praktikan perlu melakukan normalisasi fungsi kegagalan dari sistem yang ditinjau untuk setiap variabel-variabelnya. Input data yang diperlukan dalam praktikum ini adalah:

Nilai-nilai variabel yang terdapat pada fungsi kegagalan yang telah dinormalisasi A, B, C, D dan E. Harga awal untuk setiap nilai alpha dan betha. Yang perlu diperhatikan disini semua data input untuk praktikum ini (AFOSM) berbentuk real, apabila praktikan melakukan kesalahan memasukkan data, akan ditampilkan pesan kesalahan. Apabila praktikan belum mengerti format real yang berlaku didalam komputer yang sedang dipakai, dapat dilihat pada Regional Setting dalam group box Control Panel.

File Help

Input Data Hasil Program Detail Hasil

Input Panel

Basic Form: $A + B^X + C^Y + D^Z + E^WZ = D$

Real Form:

A = B = C =

D = E =

Harga awal

Alpha 1

Alpha 2

Alpha 3

Betha

OK

Gambar tampilan input untuk metode AFOSM

Pada gambar diatas terlihat form input untuk praktikum I, yang terdiri atas:

1. Baris menu



2. Baris toolbar



3. Page control dengan tiga tabsheet.

Input Data | Hasil Program | Cetak Hasil

Input Panel

Basic Form: $A + B^*X + C^*Y + D^*Z + E^*Y^*Z = 0$

Real Form:

Perintah-perintah yang dapat dilakukan pada form input ini adalah:

1. Melakukan reset data masukan.

Hal ini dapat dilakukan

- Memilih baris menu File-Baru.
- Menekan tombol Baru pada baris tool bar.



2. Membuka file data input

Membuka file data input juga dapat dilakukan dengan dua cara:

- Memilih lewat baris menu File-Buka file input.
- Menekan tombol buka pada baris tool bar.

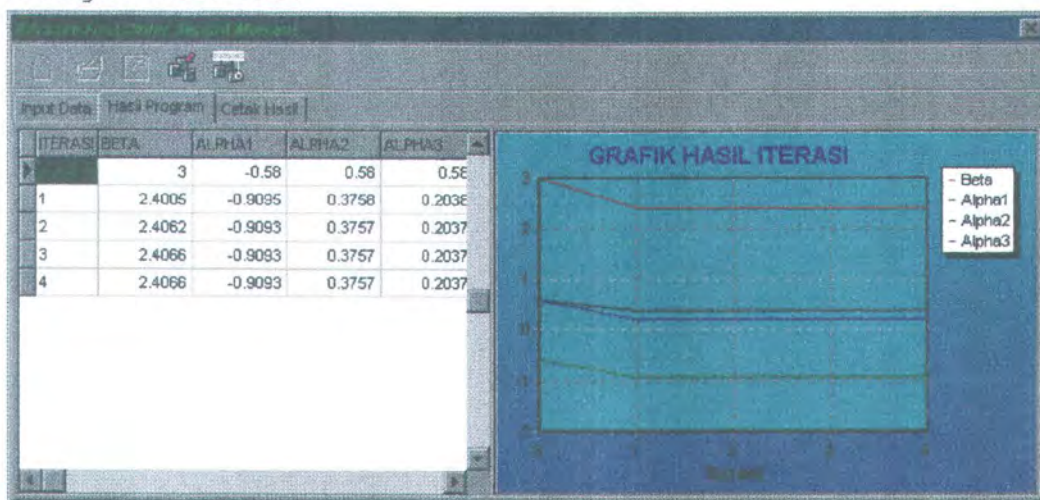


Setelah memasukkan data-data input yang diperlukan, praktikan dapat melakukan praktikum dengan menekan tombol OK. Program akan melakukan iterasi terhadap fungsi kegagalan yang diberikan, dimana jumlah iterasi tergantung pada harga nilai awal yang diberikan. Iterasi akan berlangsung cepat bila pemilihan harga awal dilakukan secara tepat.

Proses akan menghasilkan output berupa tabel dan grafik yang dapat dilihat dengan memilih tabsheet Hasil program

Input Data | Hasil Program | Cetak Hasil

Control page akan memperlihatkan tabsheet Hasil program sebagai berikut:



Gambar form tab sheet hasil program aktif.

Gambar diatas menunjukkan tampilan form AFOSM dengan tab sheet Hasil program yang aktif.

Pada gambar tersebut dapat dilihat tab sheet hasil program, yang terdiri dari tabel hasil iterasi dan grafik hasil iterasi untuk setiap harga betha dan alpha. Sampai pada langkah ini praktikan dapat menyimpan hasil program tersebut dengan cara:

- Memilih baris menu file Simpan-File output.
- Menekan tombol simpan file output pada tool bar.



Selain itu praktikan dapat menyimpan data-data input yang telah diproses dengan cara:

- Memilih baris menu file Simpan-File input.
- Menekan tombol simpan file input pada tool bar.



Langkah selanjutnya adalah memilih tabsheet Cetak hasil program.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Analisa Keandalan dan Resiko menggunakan AFOSM																	
[HASIL PRAKTIKUM]																	
Title																	
Colors Header																	
Detail																	
Iterasi																	
Beta																	
Alpha 1																	
Alpha 2																	
Alpha 3																	

Gambar tampilan form dengan tabsheet cetak hasil program sedang aktif.

Pada tampilan ini terlihat format halaman untuk laporan yang akan dicetak. Format ini sudah baku dan tidak bisa dirubah. Proses pencetakan hasil program dilakukan dengan:

- Memilih baris menu Cetak hasil iterasi.
- Menekan tombol print preview pada tab sheet Cetak hasil.



Setelah selesai melakukan praktikum program dapat dimatikan dengan memilih baris menu File-Keluar atau menekan tombol keluar pada baris tool bar.



Tampilan Pilihan program akan terlihat aktif kembali, bila praktikan ingin meneruskan praktikum dapat dilakukan pemilihan program pilihan yang lain menekan tombol Selesai bila ingin keluar dari program utama.

Praktikum II

Praktikum II adalah praktikum yang menggunakan Simulasi Monte Carlo. Sebelum praktikum dilaksanakan perlu dilakukan persiapan-persiapan sebagai berikut:

1. Menentukan jumlah percobaan.
2. Menentukan jumlah variabel acak dasar.
3. Menentukan nilai rata-rata, simpangan baku dan jenis distribusi untuk setiap variabel acak dasar yang telah ditentukan sebelumnya.

Praktikum ini dapat dilaksanakan dengan cara menekan/memilih radio button Simulasi Monte Carlo pada tampilan utama program (form pilihan program). Selanjutnya akan terlihat tampilan praktikum Simulasi Monte Carlo.

Praktikum II

File Help

Input Data Hasil Uktum Tabel dan grafik Cetak hasil

Jumlah perhitungan: 1000

Jumlah variabel acak: 3

Data Variabel Acak Dasar No-3

Nilai rata-rata:

Nilai simpangan baku:

Tipe variabel no-1

☐ Normal

☐ Log Normal

Lanjut

Ulang

BATAL

OKE

Data	Rata-rata	Simpangan Baku	Distribusi
1	1	1	Normal
2	2	2	Log Normal
3	3	3	Normal

Gambar tampilan praktikum II

Pada gambar diatas, form terbagi atas:

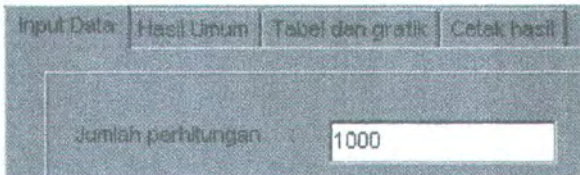
1. Baris menu.

File Help

2. Baris tool bar.



3. Control page yang terdiri dari empat tabsheet, yaitu:



- Input Data
- Hasil Umum
- Tabel dan Grafik
- Cetak Hasil

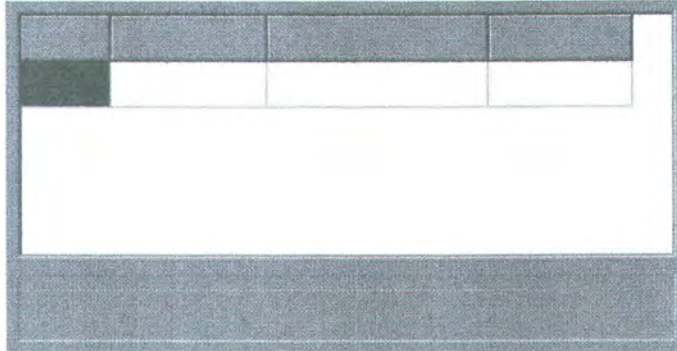
4. Tombol Ulang dan Lanjut untuk memasukkan data input. Terlihat tombol Lanjut sedang dimatikan, hal ini terjadi bila data input belum dimasukkan.



5. Tombol Batal untuk membatalkan praktikum SMC yang telah dipilih dan OKE untuk melakukan proses simulasi bila data input telah dimasukkan.



6. Tabel data input untuk mengetahui input yang telah dimasukkan.




7. Progress bar untuk menunjukkan proses simulasi sedang berlangsung.

Langkah pertama praktikum II adalah memasukkan data input, jumlah percobaan dan jumlah variabel acak. Data input untuk variabel acak dimasukkan pada bagian kiri bawah form. Data input variabel acak harus dimasukkan satu persatu, untuk data input variabel acak berikutnya dilakukan dengan menekan tombol Lanjut. Bila terjadi kesalahan penulisan data input tersebut dapat diulang dengan menekan tombol Ulang.

Data input untuk jumlah percobaan dan jumlah variabel acak adalah bertipe byte (bilangan asli), sedangkan data input nilai rata-rata dan simpangan baku untuk

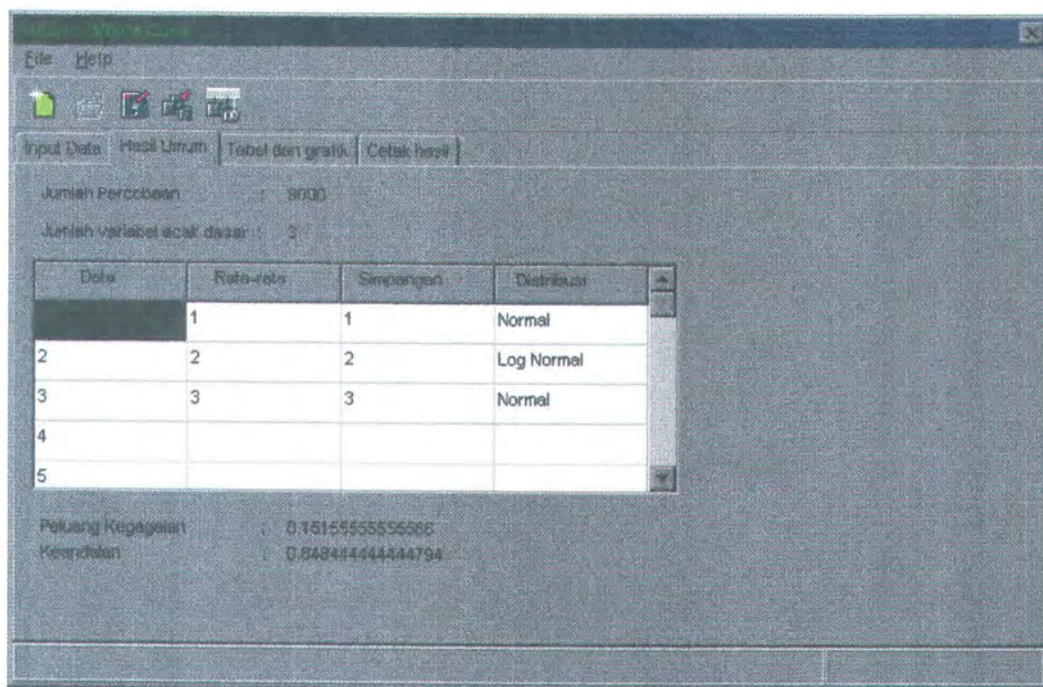
setiap variabel acak bertipe real (rasional). Bila praktikan melakukan kesalahan akan ditampilkan pesan kesalahan dan praktikan harus mengulangi/mengganti nilai data input yang salah tersebut.

Data input juga dapat diambil dari file yang telah disimpan sebelumnya dengan cara:

- Memilih baris menu File-Buka file input.
- Menekan tombol Buka pada tool bar. 

Setelah semua data input dimasukkan, proses simulasi dilakukan dengan menekan tombol OKE, selama proses berlangsung akan terlihat progress bar berjalan sesuai dengan jumlah simulasi yang dilakukan.

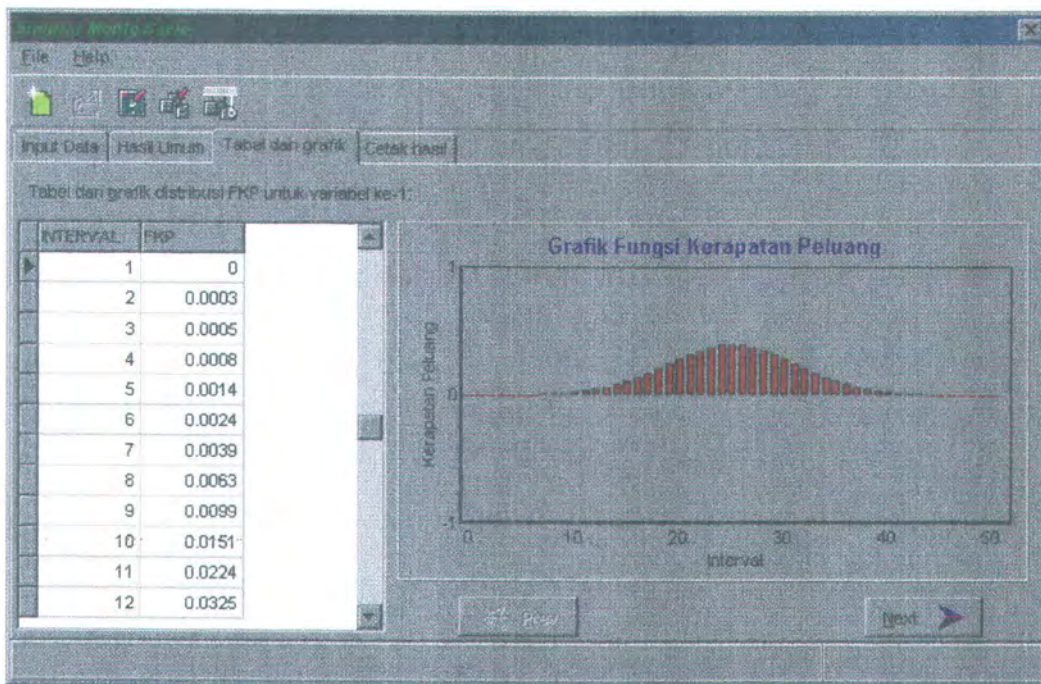
Hasil simulasi yang telah dilakukan dapat dilihat pada tab sheet Hasil umum.



Gambar tampilan hasil umum simulasi monte carlo.

Pada tampilan diatas ditunjukkan besarnya keandalan dan peluang kegagalan sistem yang analisa.

Hasil simulasi yang lebih detail terdapat pada tab sheet Tabel dan grafik.

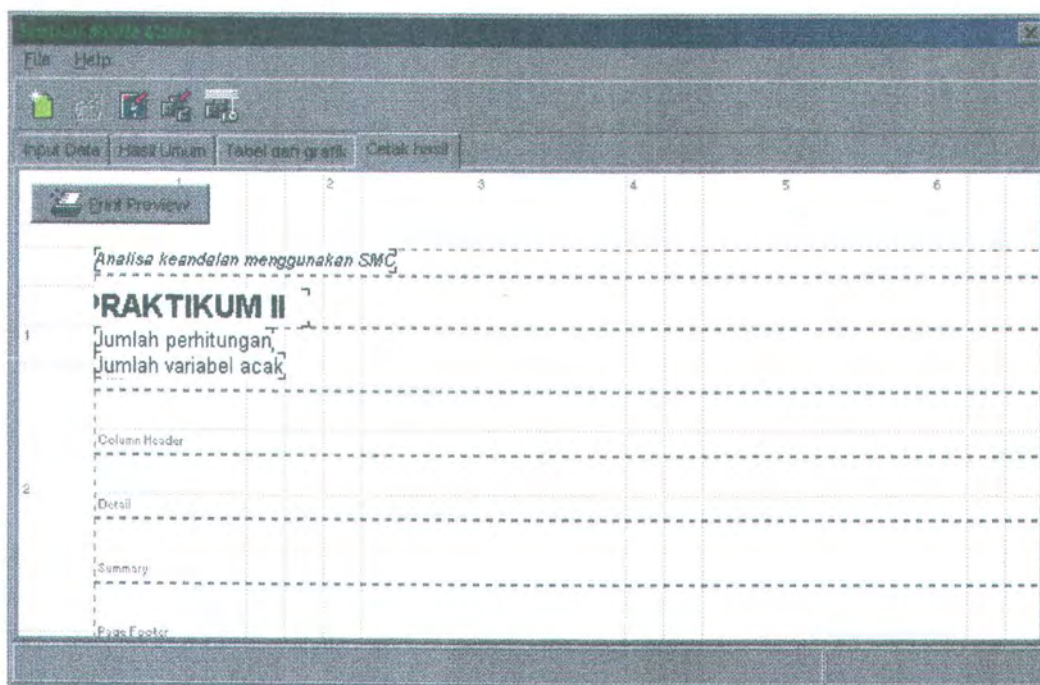


Gambar tampilan tabel dan grafik hasil simulasi.

Untuk melihat hasil tabel dan grafik setiap data variabel dilakukan dengan menekan tombol Next dan Prev. Setelah praktikum II dilaksanakan praktikan dapat menyimpan data input dengan cara:

- Memilih baris menu File-Simpan data input.
- Menekan tombol Simpan data input pada tool bar dan data output dengan cara:
- Memilih baris menu File-Simpan file output.
- Menekan tombol Simpan file output pada tool bar

Pada akhir praktikum, praktikan harus mencetak hasil praktikum untuk membuat laporan hasil praktikum. Pilih tab sheet Cetak hasil,



Gambar tampilan cetak hasil praktikum II

dan menekan tombol print preview untuk melihat tampilan pra-cetak.

Praktikum II diakhiri dengan menekan tombol Keluar pada baris toolbar.



Seperti telah dijelaskan diatas, tampilan utama akan terlihat kembali setelah setiap praktikum diakhiri.



LAMPIRAN B LISTING PROGRAM

Maka sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kepayahan (jasmani) itu ada pula kelapangan. Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah urusan yang lain dengan sungguh-sungguh. (QS. Al Insyirah: 5-7)

```
1: unit Unit1;
2:
3: interface
4:
5: uses
6:   SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
7:   StdCtrls, ExtCtrls, Forms, Buttons;
8:
9: type
10:   TPilProg = class(TForm)
11:     Batal: TBitBtn;
12:     RadioGroup1: TRadioGroup;
13:     Advanced: TRadioButton;
14:     FTA: TRadioButton;
15:     MonteCarlo: TRadioButton;
16:     BitBtn1: TBitBtn;
17:     procedure MonteCarloClick(Sender: TObject);
18:     procedure BatalClick(Sender: TObject);
19:     procedure BitBtn1Click(Sender: TObject);
20:     procedure AdvancedClick(Sender: TObject);
21:
22:   end;
23:
24: var
25:   PilProg: TPilProg;
26:
27: implementation
28: uses Unit2, Unit3, Unit4;
29: {$R *.DFM}
30:
31: procedure TPilProg.MonteCarloClick(Sender: TObject);
32: begin
33:   SMC:= TSMC.Create(Self);
34:   SMC.ShowModal;
35: {   PilProg.Close;}
36: end;
37:
38: procedure TPilProg.BatalClick(Sender: TObject);
39: begin
40: Close;
41: end;
42:
43:
44: procedure TPilProg.BitBtn1Click(Sender: TObject);
45: begin
46: AboutBox.Show;
47: end;
48:
49: procedure TPilProg.AdvancedClick(Sender: TObject);
50: begin
51:   AFOSM:= TAFOSM.Create(Self);
52:   AFOSM.ShowModal;
53: {   PilProg.Close;}
54: end;
55:
56: end.
```



```
1: unit Unit2;
2:
3: interface
4:
5: uses WinTypes, WinProcs, Classes, Graphics, Forms, Controls, StdCtrls,
6:   Buttons, ExtCtrls;
7:
8: type
9:   TAboutBox = class(TForm)
10:     Panell1: TPanel;
11:     OKButton: TBitBtn;
12:     ProgramIcon: TImage;
13:     ProductName: TLabel;
14:     Copyright: TLabel;
15:     Panel2: TPanel;
16:     Label2: TLabel;
17:     Label1: TLabel;
18:     Label4: TLabel;
19:     Label3: TLabel;
20:     Label5: TLabel;
21:     procedure OKButtonClick(Sender: TObject);
22:   end;
23:
24: var
25:   AboutBox: TAboutBox;
26:
27: implementation
28: uses Unit1;
29: {$R *.DFM}
30:
31: procedure TAboutBox.OKButtonClick(Sender: TObject);
32: {var PilProg: TPilProg;}
33: begin
34:   Close;
35: end;
36:
37: end.
38:
```



```

1: unit Unit3;
2:
3: interface
4:
5: uses
6:   SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
7:   Forms, Dialogs, StdCtrls, ExtCtrls, Buttons, Hitung, Menus, ComCtrls,
8:   Grids, TeEngine, Series, TeeProcs, Chart, Db, DBTables, DBGrids,
9:   DBChart, quickrpt, QrCtrls; { VBXCtrl, Gauge;}
10:
11: type
12:   TSMC = class(TForm)
13:     Panel2: TPanel;
14:     Panel4: TPanel;
15:     SpeedButton1: TSpeedButton;
16:     SpeedButton2: TSpeedButton;
17:     SpeedButton3: TSpeedButton;
18:     Panel5: TPanel;
19:     Panel6: TPanel;
20:     MainMenu1: TMainMenu;
21:     OpenDialog2: TOpenDialog;
22:     Berkas1: TMenuItem;
23:     Barul: TMenuItem;
24:     Bukal: TMenuItem;
25:     Simpan1: TMenuItem;
26:     Keluar1: TMenuItem;
27:     N2: TMenuItem;
28:     Help1: TMenuItem;
29:     Infol: TMenuItem;
30:     PageControl1: TPageControl;
31:     TabSheet1: TTabSheet;
32:     TabSheet2: TTabSheet;
33:     TabSheet3: TTabSheet;
34:     TabSheet4: TTabSheet;
35:     GroupBox1: TGroupBox;
36:     RadioGroup1: TRadioGroup;
37:     Lanjut: TButton;
38:     Ulang: TButton;
39:     ProgressBar1: TProgressBar;
40:     Oke: TBitBtn;
41:     Batal: TBitBtn;
42:     SaveDialog1: TSaveDialog;
43:     SaveDialog2: TSaveDialog;
44:     SaveDialog3: TSaveDialog;
45:     NilsBaku: TEdit;
46:     Label3: TLabel;
47:     Label4: TLabel;
48:     Normal: TRadioButton;
49:     LogNormal: TRadioButton;
50:     StringGrid1: TStringGrid;
51:     Panel1: TPanel;
52:     Label1: TLabel;
53:     Label2: TLabel;
54:     JumHit: TEdit;
55:     JumVar: TEdit;
56:     NilRat: TEdit;
57:     Label7: TLabel;
58:     Label8: TLabel;
59:     StringGrid2: TStringGrid;
60:     Label9: TLabel;
61:     Label10: TLabel;
62:     Label11: TLabel;
63:     prev: TBitBtn;
64:     next: TBitBtn;
65:     Label12: TLabel;
66:     Label13: TLabel;

```

```

67:   Label14: TLabel;
68:   Label15: TLabel;
69:   SaveDialog4: TSaveDialog;
70:   DataSource1: TDataSource;
71:   Table2: TTable;
72:   Table1: TTable;
73:   DBGrid1: TDBGrid;
74:   DBChart1: TDBChart;
75:   Series1: TBarSeries;
76:   InputSMC: TMenuItem;
77:   OutputSMC: TMenuItem;
78:   SpeedButton4: TSpeedButton;
79:   SpeedButton5: TSpeedButton;
80:   QuickRep2: TQuickRep;
81:   PageHeaderBand1: TQRBand;
82:   TitleBand1: TQRBand;
83:   ColumnHeaderBand1: TQRBand;
84:   DetailBand1: TQRBand;
85:   SummaryBand1: TQRBand;
86:   PageFooterBand1: TQRBand;
87:   QRLabel1: TQRLabel;
88:   QRLabel2: TQRLabel;
89:   QRBand1: TQRBand;
90:   QRLabel3: TQRLabel;
91:   QRLabel4: TQRLabel;
92:   BitBtn1: TBitBtn;
93:   SaveDialog5: TSaveDialog;
94:   {procedure ShowHint(Sender:TObject);}
95:   procedure BatalClick(Sender: TObject);
96:   procedure JumVarChange(Sender: TObject);
97:   procedure LanjutClick(Sender: TObject);
98:   procedure NormalClick(Sender: TObject);
99:   procedure LogNormalClick(Sender: TObject);
100:  procedure JumHitChange(Sender: TObject);
101:  procedure NilRatChange(Sender: TObject);
102:  procedure NilsBakuChange(Sender: TObject);
103:  procedure OkeClick(Sender: TObject);
104:  procedure BarulClick(Sender: TObject);
105:  procedure BukalClick(Sender: TObject);
106:  procedure Simpan1Click(Sender: TObject);
107:  procedure Keluar1Click(Sender: TObject);
108:  procedure Tutup2Click(Sender: TObject);
109:  procedure Tumpuk1Click(Sender: TObject);
110:  procedure Jajar1Click(Sender: TObject);
111:  procedure SpeedButton3Click(Sender: TObject);
112:  procedure SpeedButton2Click(Sender: TObject);
113:  procedure SpeedButton1Click(Sender: TObject);
114:  procedure InfolClick(Sender: TObject);
115:  procedure UlangClick(Sender: TObject);
116:  procedure FormCreate(Sender: TObject);
117:  procedure prevClick(Sender: TObject);
118:  procedure nextClick(Sender: TObject);
119:  procedure SpeedButton4Click(Sender: TObject);
120:  procedure SpeedButton5Click(Sender: TObject);
121:  procedure InputSMCClick(Sender: TObject);
122:  procedure OutputSMCClick(Sender: TObject);
123:  procedure BitBtn1Click(Sender: TObject);
124:
125:  private
126:    { Private declarations }
127:  public
128:    { Public declarations }
129:  end;
130:
131:  var
132:    SMC: TSMC;

```

```

133:   NewTable           : array [1..10] of TTable;
134:   UpDateTable2       : TTable;
135:   MyDataSource        : TDataSource;
136:   FirstTime           : boolean;
137:
138: implementation
139: {$R *.DFM}
140: uses Unit1, Unit2;
141: var
142:   y, z: byte;
143:   x: integer;
144: var
145:   InputPraktikum2      : textfile;
146:   NamaFile2, IsiData2  : string;
147:
148: Procedure CreateTableBuff;
149: Begin
150:   With SMC.Table1 Do Begin
151:     Active:=False;
152:     TableName := 'c:\Temporary.dbf';
153:     FieldDefs.Clear;
154:     FieldDefs.Add('Interval', ftInteger, 0, False);
155:     FieldDefs.Add('FKP', ftFloat, 0, False);
156:     CreateTable;
157:   End;
158: End;
159:
160: procedure TSMC.BatalClick(Sender: TObject);
161: begin
162:   Unit1.PilProg.Show;
163:   Close;
164:   PilProg.MonteCarlo.Checked:=False;
165:   {Dispose(Hasil);}
166: end;
167:
168: procedure TSMC.JumVarChange(Sender: TObject);
169: begin
170:   J:=0;
171:   if
172:     (Jumhit.Text<>'') and
173:     (Jumvar.Text<>'') and
174:     (Nilrat.Text<>'') and
175:     (Nilsbaku.Text<>'') and
176:     ((Normal.checked=true) or (Lognormal.checked=true))
177:   then
178:     begin
179:       Lanjut.Enabled:=True;
180:       Panel4.Tag:=0;
181:     end;
182:     GroupBox1.Caption:='Data variabel acak dasar ke-1: ';
183:     RadioGroup1.Caption:='Tipe variabel ke-1';
184:   end;
185:
186: procedure TSMC.LanjutClick(Sender: TObject);
187:
188: begin
189:   Lanjut.enabled:=false;
190:   J:=J+1;
191:   JumVar.enabled:=false;
192:   StringGrid1.RowCount:= StringGrid1.RowCount+1;
193:   Try
194:     Ntrial:=StrToInt(JumHit.Text);
195:   except
196:     on EConvertError do
197:       begin
198:         MessageDlg('Nilai'+JumHit.Text+'Dalam "Jumlah Perhitungan" Bukan Integer,
```



```

        coba lagi ....!!!',MtWarning, [mbOK],0);
199: JumHit.Enabled:=True;
200: JumHit.Focused;
201: end;
202: end;
203:
204:
205: Try
206: Nranvr:=StrToInt(JumVar.Text);
207: except
208: on EConvertError do
209: begin
210: MessageDlg('Nilai'+JumVar.Text+'Dalam "Jumlah Variabel Acak Dasar" Bukan
211: Integer, coba lagi ....!!!',MtWarning, [mbOK],0);
212: JumVar.Enabled:=True;
213: JumVar.Focused;
214: end;
215: end;
216:
217:
218: With VarAcak[J] do
219:     begin
220:
221:         Try
222:         NilaiRata:=StrToFloat(NilRat.Text);
223:         except
224:         on EConvertError do
225:         begin
226:         MessageDlg('Nilai'+NilRat.Text+'Dalam "Nilai rata-rata" Bukan Real,
227:             coba lagi ....!!!',MtWarning, [mbOK],0);
228:         NilRat.Enabled:=True;
229:         NilRat.Focused;
230:         end;
231:         end;
232:         StringGrid1.Cells[0,J]:= IntToStr(J);
233:         StringGrid1.Cells[1,J]:= NilRat.Text;
234:         StringGrid2.Cells[0,J]:= IntToStr(J);
235:         StringGrid2.Cells[1,J]:= NilRat.Text;
236:
237:         Try
238:         NilaiSBaku:=StrToFloat(NilSBaku.Text);
239:         except
240:         on EConvertError do
241:         begin
242:         MessageDlg('Nilai'+NilSBaku.Text+'Dalam "Nilai Simpangan Baku" Bukan
243:             Real, coba lagi ....!!!',MtWarning, [mbOK],0);
244:         NilSBaku.Enabled:=True;
245:         NilSBaku.Focused;
246:         end;
247:         end;
248:         StringGrid1.Cells[2,J]:= NilSBaku.Text;
249:         StringGrid2.Cells[2,J]:= NilSBaku.Text;
250:
251:         if Normal.Checked = true then
252:         begin
253:         Distribusi:=true;
254:         StringGrid1.Cells[3,J]:= 'Normal';
255:         StringGrid2.Cells[3,J]:= 'Normal';
256:         end else
257:         begin
258:         StringGrid1.Cells[3,J]:= 'Log Normal';
259:         StringGrid2.Cells[3,J]:= 'Log Normal';
260:         end;
261:         end;
262:
263: GroupBox1.Caption:='Data Variabel Acak Dasar ke-'+IntToStr(J)+' :';

```

```
264: NilRat.Text:='';
265: NilsBaku.Text:='';
266: NilRat.Enabled:=True;
267: Ulang.Enabled:=True;
268: Normal.Checked:=False;
269: if (J= Nranvr) then
270: begin
271: GroupBox1.Caption:='Data Variabel Acak Dasar ke-'+IntToStr(J)+' :';
272: Lanjut.Enabled:= false;
273: Oke.Enabled:= true;
274: end;
275: end;
276:
277: procedure TSMC.NormalClick(Sender: TObject);
278: begin
279: if
280: (Jumhit.Text<>'')and
281: (Jumvar.Text<>'')and
282: (Nilrat.Text<>'')and
283: (Nilsbaku.Text<>'')and
284: ((Normal.checked=true) or (Lognormal.checked=true))
285: then
286: begin
287: Lanjut.Enabled:=True;
288: Panel4.Tag:=0;
289: end;
290: end;
291:
292: procedure TSMC.LogNormalClick(Sender: TObject);
293: begin
294: if
295: (Jumhit.Text<>'')and
296: (Jumvar.Text<>'')and
297: (Nilrat.Text<>'')and
298: (Nilsbaku.Text<>'')and
299: ((Normal.checked=true) or (Lognormal.checked=true))
300: then
301: begin
302: Lanjut.Enabled:=True;
303: Panel4.Tag:=0;
304: end;
305: end;
306:
307: procedure TSMC.JumHitChange(Sender: TObject);
308: begin
309: if
310: (Jumhit.Text<>'')and
311: (Jumvar.Text<>'')and
312: (Nilrat.Text<>'')and
313: (Nilsbaku.Text<>'')and
314: ((Normal.checked=true) or (Lognormal.checked=true))
315: then
316: begin
317: Lanjut.Enabled:=True;
318: Panel4.Tag:=0;
319: end;
320: {JumVar.Enabled:= True;}
321: end;
322:
323: procedure TSMC.NilRatChange(Sender: TObject);
324: begin
325: if
326: (Jumhit.Text<>'')and
327: (Jumvar.Text<>'')and
328: (Nilrat.Text<>'')and
329: (Nilsbaku.Text<>'')and
```

```
330: ((Normal.checked=true) or (Lognormal.checked=true))
331: then
332: begin
333: Lanjut.Enabled:=True;
334: Panel4.Tag:=0;
335: end;
336: end;
337:
338: procedure TSMC.NilsBakuChange(Sender: TObject);
339: begin
340: if
341: (Jumhit.Text<>'')and
342: (Jumvar.Text<>'')and
343: (Nilrat.Text<>'')and
344: (Nilsbaku.Text<>'')and
345: ((Normal.checked=true) or (Lognormal.checked=true))
346: then
347: begin
348: Lanjut.Enabled:=True;
349: Panel4.Tag:=0;
350: end;
351: end;
352:
353: procedure TSMC.OkeClick(Sender: TObject);
354: begin
355: if not FirstTime then begin
356: CreateTableBuff;
357: FirstTime:=true;
358: end;
359: SMC.Table2.Active:=false;
360: Utama;
361: PageControl1.ActivePage:=TabSheet2;
362: z:=2;
363: StringGrid2.Cells[0,0]:= '      Data';
364: StringGrid2.Cells[1,0]:= '      Rata-rata';
365: StringGrid2.Cells[2,0]:= '      Simpangan';
366: StringGrid2.Cells[3,0]:= '      Distribusi';
367: {StringGrid2.Cells[0,0]:= 'Interval';
368: StringGrid2.Cells[1,0]:= 'FKP';}
369: for y:=1 to 51 do
370: StringGrid2.Cells[0,y]:= IntToStr(y);
371: end;
372:
373: procedure TSMC.BarulClick(Sender: TObject);
374: begin
375: OutputSMC.Enabled:=False;
376: SpeedButton1.Enabled:=False;
377: JumVar.enabled:=false;
378: Panel4.Tag:=1;
379: PageControl1.Tag:=0;
380: Jumhit.Text:='';
381: Jumvar.Text:='';
382: Nilrat.Text:='';
383: Nilsbaku.Text:='';
384: Normal.checked:=False;
385: Lognormal.checked:=False;
386: StringGrid1.DestroyComponents;
387: StringGrid1.RowCount:=2;
388: Lanjut.Enabled:=False;
389: Ulang.Enabled:=False;
390: Oke.Enabled:=False;
391: end;
392:
393: procedure TSMC.Buka1Click(Sender: TObject);
394: var
395: BK, JumlahData2: Integer;
```



```
396: begin
397: Ulang.Enabled:=false;
398: OpenFileDialog2.FileName:='';
399: if OpenFileDialog2.Execute then
400: begin
401: NamaFile2:=OpenDialog2.FileName;
402: try
403: AssignFile(InputPraktikum2,NamaFile2);
404: reset(InputPraktikum2);
405: finally
406: with SMC do
407: begin
408: Jumvar.enabled:=true;
409: readln(InputPraktikum2,IsiData2);
410: Jumhit.Text:=IsiData2;
411: readln(InputPraktikum2,IsiData2);
412: Jumvar.Text:=IsiData2;
413: JumlahData2:=StrToInt(Jumvar.Text);
414: for BK:=1 to JumlahData2 do
415: begin
416: readln(InputPraktikum2,IsiData2);
417: Nilrat.Text:=IsiData2;
418: readln(InputPraktikum2,IsiData2);
419: Nilsbaku.Text:=IsiData2;
420: readln(InputPraktikum2,IsiData2);
421: if IsiData2='Normal' then
422: Normal.checked:=true
423: else LogNormal.checked:=true;
424: Lanjut.Click;
425: end;
426: end;
427: CloseFile(InputPraktikum2);
428: end;
429: end;
430: end;
431:
432: procedure TSMC.Simpan1Click(Sender: TObject);
433: begin
434: { save current file (ActiveMDIChild points to the window) }
435: end;
436:
437: procedure TSMC.Keluar1Click(Sender: TObject);
438: begin
439: Close;
440: Pilprog.MonteCarlo.Checked:=false;
441: end;
442:
443: procedure TSMC.Tutup2Click(Sender: TObject);
444: begin
445: if ActiveMDIChild <> nil then
446: ActiveMDIChild.Close;
447:
448: end;
449:
450: procedure TSMC.Tumpuk1Click(Sender: TObject);
451: begin
452: Cascade;
453: end;
454:
455: procedure TSMC.Jajar1Click(Sender: TObject);
456: begin
457: Tile;
458: end;
459:
460: procedure TSMC.SpeedButton3Click(Sender: TObject);
461: var
```

```
462:     a: integer;
463: begin
464:     Close;
465:     PilProg.Batal.Caption:='&Selesai';
466:     PilProg.MonteCarlo.Checked:=false;
467:     {   for a:=1 to 10 do
468:         begin
469:             With NewTable[a] do
470:                 try
471:                     NewTable[a].EmptyTable;
472:                 finally
473:                     end;
474:                 end;
475:         }
476:     end;
477:
478: procedure TSMC.SpeedButton2Click(Sender: TObject);
479: begin
480:     InputSMC.Click;
481: end;
482:
483: procedure TSMC.SpeedButton1Click(Sender: TObject);
484: begin
485:     Buka1.Click;
486: end;
487:
488: procedure TSMC.Info1Click(Sender: TObject);
489: begin
490:     Unit2.Aboutbox.ShowModal;
491: end;
492:
493: procedure TSMC.UlangClick(Sender: TObject);
494: begin
495:     J:=J-1;
496:     StringGrid1.RowCount:= StringGrid1.RowCount-1;
497:     NilRat.Text:='';
498:     NilSBaku.Text:='';
499:     Normal.Checked:=False;
500:     Lognormal.Checked:=False;
501:     GroupBox1.Caption:='Data Variabel Acak Dasar ke -'+IntToStr(J+1)+':';
502:     if J=0 then
503:     begin
504:         JumHit.Text:='';
505:         JumVar.Text:='';
506:         GroupBox1.Caption:='';
507:         Lanjut.Enabled:=False;
508:     end;
509: end;
510: procedure TSMC.FormCreate(Sender: TObject);
511: var
512:     a:integer;
513: begin
514:     PageControl1.ActivePage:=TabSheet1;
515:     StringGrid1.Cells[0,0]:= '   Data';
516:     StringGrid1.Cells[1,0]:= '   Rata-rata';
517:     StringGrid1.Cells[2,0]:= '   Simpangan Baku';
518:     StringGrid1.Cells[3,0]:= '   Distribusi';
519:
520:     FirstTime:=False;
521:     for a:=1 to 10 do
522:     begin
523:         NewTable[a]:=TTable.Create(Self);
524:         With NewTable[a] do
525:             Begin
526:                 Active:=False;
527:                 DatabaseName := 'DBDEMOS';
```

```
528:     TableName := 'c:\bufer'+Inttostr(a)+'.dbf';
529:     TableType := ttDefault;
530:     FieldDefs.Clear;
531:     FieldDefs.Add('Interval', ftInteger, 0, False);
532:     FieldDefs.Add('FKP', ftFloat, 0, False);
533:     CreateTable;
534:     NewTable[a].EmptyTable;
535: end;
536: end;
537:
538: end;
539: procedure TSMC.prevClick(Sender: TObject);
540: begin
541:     Next.Enabled:=true;
542:     DBGrid1.Tag:=DBGrid1.Tag-1;
543:     if DBGrid1.Tag=1
544:     then Prev.Enabled:=false;
545:     SMC.Table2.Active:=false;
546:     SMC.Table2.TableName:='c:\bufer'+IntToStr(DBGrid1.Tag)+'.dbf';
547:     SMC.Table2.Active:=true;
548:     Label11.Caption:='Tabel dan grafik distribusi FKP untuk variabel
549:         ke-'+IntToStr(DBGrid1.Tag);
550: end;
551:
552: procedure TSMC.nextClick(Sender: TObject);
553: begin
554:     Prev.Enabled:=true;
555:     DBGrid1.Tag:=DBGrid1.Tag+1;
556:     if DBGrid1.Tag= Nranvr
557:     then Next.Enabled:=false;
558:     SMC.Table2.Active:=false;
559:     SMC.Table2.TableName:='c:\bufer'+IntToStr(DBGrid1.Tag)+'.dbf';
560:     SMC.Table2.Active:=true;
561:     Label11.Caption:='Tabel dan grafik distribusi FKP untuk variabel
562:         ke-'+IntToStr(DBGrid1.Tag);
563: end;
564:
565: procedure TSMC.SpeedButton4Click(Sender: TObject);
566: begin
567: Barul.Click;
568: end;
569:
570: procedure TSMC.SpeedButton5Click(Sender: TObject);
571: begin
572: OutputSMC.Click;
573: end;
574:
575: procedure TSMC.InputSMCClick(Sender: TObject);
576: var
577: BarisTabel, KolomTabel: Integer;
578: begin
579: SaveDialog4.FileName:='';
580: SaveDialog4.DefaultExt:='edi';
581: if SaveDialog4.Execute then
582: begin
583: Namafile2:=SaveDialog4.FileName;
584: assignfile(InputPraktikum2, namafile2);
585: rewrite(InputPraktikum2);
586: writeln(InputPraktikum2, Jumhit.Text);
587: writeln(InputPraktikum2, Jumvar.Text);
588: for BarisTabel:=1 to (StringGrid1.RowCount-2) do
589: for KolomTabel:=1 to 3 do
590: writeln(InputPraktikum2, StringGrid1.Cells[KolomTabel, BarisTabel]);
591: Closefile(InputPraktikum2);
592: end;
593: end;
```



```
594:
595: procedure TSMC.OutputSMCClick(Sender: TObject);
596: begin
597:   if SaveDialog5.Execute then begin
598:     SaveDialog5.FileName:='';
599:     UpDateTable2:=TTable.Create(Self);
600:     with UpDateTable2 do
601:       begin
602:         Active:=False;
603:         DatabaseName := 'DBDEMOS';
604:         TableName := SaveDialog2.FileName;
605:         TableType := ttDefault;
606:         FieldDefs.Clear;
607:         FieldDefs.Add('Interval', ftInteger, 0, False);
608:         FieldDefs.Add('FKP', ftFloat, 0, False);
609:         CreateTable;
610:       end;
611:   {      UpDateTable2.BatchMove(SMC.Table1,batAppend);}
612:   end;
613:
614:
615: end;
616:
617: procedure TSMC.BitBtn1Click(Sender: TObject);
618: begin
619:   QuickRep2.Preview;
620: end;
621:
622: end.
623:
624:
```

```
1: unit unit4;
2:
3: interface
4:
5: uses
6:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
7:   Dialogs, StdCtrls, DBTables, Db, Grids, DBGrids, ExtCtrls,
8:   TeeProcs, TeEngine, Chart, DBChart, Series, TeeFunci, Menus,
9:   ToolWin, ComCtrls, quickrpt, QrCtrls, Buttons, QrTee;
10:
11: type
12:   String40 = string[40];
13:   TAFOSM = class(TForm)
14:     PageControl1: TPageControl;
15:     TabSheet1: TTabSheet;
16:     TabSheet2: TTabSheet;
17:     TabSheet3: TTabSheet;
18:     CoolBar1: TCoolBar;
19:     GroupBox1: TGroupBox;
20:     Label1: TLabel;
21:     Label2: TLabel;
22:     Label4: TLabel;
23:     Label5: TLabel;
24:     Label6: TLabel;
25:     Label7: TLabel;
26:     Label8: TLabel;
27:     Beta: TEdit;
28:     Alpha3: TEdit;
29:     Alpha1: TEdit;
30:     Alpha4: TEdit;
31:     Alpha2: TEdit;
32:     Edit1: TEdit;
33:     DataSourceOut: TDataSource;
34:     TableOut: TTable;
35:     MainMenu1: TMainMenu;
36:     File1: TMenuItem;
37:     New1: TMenuItem;
38:     Save1: TMenuItem;
39:     Exit1: TMenuItem;
40:     Help1: TMenuItem;
41:     About1: TMenuItem;
42:     Help2: TMenuItem;
43:     SaveDialog1: TSaveDialog;
44:     TempTable: TTable;
45:     DataTable: TDBGrid;
46:     GridOut: TDBChart;
47:     Series1: TLineSeries;
48:     Series2: TLineSeries;
49:     Series3: TLineSeries;
50:     Series4: TLineSeries;
51:     GroupBox2: TGroupBox;
52:     Label9: TLabel;
53:     IBeta: TEdit;
54:     Label10: TLabel;
55:     IAlpha1: TEdit;
56:     Label12: TLabel;
57:     Label11: TLabel;
58:     IAlpha2: TEdit;
59:     IAlpha3: TEdit;
60:     Oke: TButton;
61:     CoolBar2: TCoolBar;
62:     Preview: TSpeedButton;
63:     QuickRep1: TQuickRep;
64:     DetailBand1: TQRBand;
65:     PageFooterBand1: TQRBand;
66:     TitleBand1: TQRBand;
```

```

67:   QRLabel1: TQRLabel;
68:   QRDBText1: TQRDBText;
69:   QRDBText2: TQRDBText;
70:   QRDBText3: TQRDBText;
71:   QRDBText4: TQRDBText;
72:   QRDBText5: TQRDBText;
73:   PageHeaderBand1: TQRBand;
74:   QRLabel8: TQRLabel;
75:   QRSysData1: TQRSysData;
76:   QRLabel9: TQRLabel;
77:   QRSysData2: TQRSysData;
78:   QRLabel10: TQRLabel;
79:   SpeedButton2: TSpeedButton;
80:   SpeedButton3: TSpeedButton;
81:   SpeedButton4: TSpeedButton;
82:   QRBand1: TQRBand;
83:   QRLabel2: TQRLabel;
84:   QRLabel7: TQRLabel;
85:   QRLabel3: TQRLabel;
86:   QRLabel4: TQRLabel;
87:   QRLabel5: TQRLabel;
88:   QRBand2: TQRBand;
89:   QRDBChart1: TQRDBChart;
90:   PrintGraph: TQRChart;
91:   Series6: TLineSeries;
92:   Series7: TLineSeries;
93:   Series8: TLineSeries;
94:   Series5: TLineSeries;
95:   SpeedButton1: TSpeedButton;
96:   InputAFOSM: TMenuItem;
97:   OutputAFOSM: TMenuItem;
98:   SaveDialog2: TSaveDialog;
99:   BukaFileInput1: TMenuItem;
100:  OpenDialog1: TOpenDialog;
101:  SpeedButton5: TSpeedButton;
102:  N1: TMenuItem;
103:  procedure OkeClick(Sender: TObject);
104:  procedure FormCreate(Sender: TObject);
105:  procedure Exit1Click(Sender: TObject);
106:  procedure PreviewClick(Sender: TObject);
107:  procedure SpeedButton4Click(Sender: TObject);
108:  procedure SpeedButton3Click(Sender: TObject);
109:  procedure SpeedButton2Click(Sender: TObject);
110:  procedure New1Click(Sender: TObject);
111:  procedure IAlpha1Change(Sender: TObject);
112:  procedure IAlpha2Change(Sender: TObject);
113:  procedure IAlpha3Change(Sender: TObject);
114:  procedure IBetaChange(Sender: TObject);
115:  procedure BetaChange(Sender: TObject);
116:  procedure Alpha3Change(Sender: TObject);
117:  procedure Alpha1Change(Sender: TObject);
118:  procedure Alpha4Change(Sender: TObject);
119:  procedure Alpha2Change(Sender: TObject);
120:  procedure PageControl1Change(Sender: TObject);
121:  procedure OutputAFOSMClick(Sender: TObject);
122:  procedure SpeedButton1Click(Sender: TObject);
123:  procedure InputAFOSMClick(Sender: TObject);
124:  procedure BukaFileInput1Click(Sender: TObject);
125:  procedure SpeedButton5Click(Sender: TObject);
126:  private
127:    { Private declarations }
128:  public
129:    { Public declarations }
130:  end;
131:
132:  var

```



```

133:   AFOSM                      : TAFOSM;
134:   NewTable, UpDateTable      : TTable;
135:   MyDataSource                : TDataSource;
136:   RealEq                      : string[40];
137:   Al, IAl                    : array[1..5] of Real;
138:   IStr                        : array[1..5] of String;
139:   I                            : integer;
140:   FirstTime                   : boolean;
141:   TAwal                       : real;
142:
143: implementation
144:
145: uses Unit1;
146:
147: var
148:   InputPraktikum1             : textfile;
149:   j, k, l, m                  : integer;
150:   NamaFile, IsiData           : string;
151:
152:
153: {$R *.DFM}
154:
155: procedure CreateTableBuff;
156: begin
157:     with AFOSM.TempTable do begin
158:         Active:=False;
159:         TableName := 'c:\Tempre.dbf';
160:         FieldDefs.Clear;
161:         FieldDefs.Add('Iterasi', ftString, 4, False);
162:         FieldDefs.Add('Beta', ftFloat, 0, False);
163:         FieldDefs.Add('Alpha1', ftFloat, 0, False);
164:         FieldDefs.Add('Alpha2', ftFloat, 0, False);
165:         FieldDefs.Add('Alpha3', ftFloat, 0, False);
166:         CreateTable;
167:     end;
168: end;
169:
170:
171: procedure Initialize;
172: var
173:   I          : Integer;
174: begin
175:   RealEq:='A+BX+CY+DZ+EYZ=0';
176:   TAwal:=0;
177:   for I:=1 to 5 do
178:       Al[I]:=0;
179:       with AFOSM do
180:           begin
181:               try
182:                   Al[1]:=StrToFloat(Beta.Text);
183:               except on EConvertError do
184:                   begin
185:                       MessageDlg('Nilai masukan "A" harus berbentuk
real', mtWarning, [mbOK], 0);
186:                       Oke.Enabled:=False;
187:                   end;
188:                   end;
189:               try
190:                   Al[2]:=StrToFloat(Alpha1.Text);
191:               except on EConvertError do
192:                   begin
193:                       MessageDlg('Nilai masukan "B" harus berbentuk
real', mtWarning, [mbOK], 0);
194:                       Oke.Enabled:=False;
195:                   end;
196:                   end;

```

```
197:         try
198:         Al[3]:=StrToFloat(Alpha2.Text);
199:     except on EConvertError do
200:     begin
201:         MessageDlg('Nilai masukan "C" harus berbentuk
real',mtWarning,[mbOK],0);
202:         Oke.Enabled:=False;
203:     end;
204:     end;
205:     try
206:         Al[4]:=StrToFloat(Alpha3.Text);
207:     except on EConvertError do
208:     begin
209:         MessageDlg('Nilai masukan "D" harus berbentuk real',
210:             mtWarning,[mbOK],0);
211:         Oke.Enabled:=False;
212:     end;
213:     end;
214:     try
215:         Al[5]:=StrToFloat(Alpha4.Text);
216:     except on EConvertError do
217:     begin
218:         MessageDlg('Nilai masukan "E" harus berbentuk real',
219:             mtWarning,[mbOK],0);
220:         Oke.Enabled:=False;
221:     end;
222:     end;
223:     try
224:         IAl[1]:=StrToFloat(IBeta.Text);
225:     except on EConvertError do
226:     begin
227:         MessageDlg('Nilai masukan "BETHA" harus berbentuk real',
228:             mtWarning,[mbOK],0);
229:         Oke.Enabled:=False;
230:     end;
231:     end;
232:     try
233:         IAl[2]:=StrToFloat(IAAlpha1.Text);
234:     except on EConvertError do
235:     begin
236:         MessageDlg('Nilai masukan "ALPHA 1" harus berbentuk real',
237:             mtWarning,[mbOK],0);
238:         Oke.Enabled:=False;
239:     end;
240:     end;
241:     try
242:         IAl[3]:=StrToFloat(IAAlpha2.Text);
243:     except on EConvertError do
244:     begin
245:         MessageDlg('Nilai masukan "ALPHA 2" harus berbentuk real',
246:             mtWarning,[mbOK],0);
247:         Oke.Enabled:=False;
248:     end;
249:     end;
250:     try
251:         IAl[4]:=StrToFloat(IAAlpha3.Text);
252:     except on EConvertError do
253:     begin
254:         MessageDlg('Nilai masukan "ALPHA 3" harus berbentuk real',
255:             mtWarning,[mbOK],0);
256:         Oke.Enabled:=False;
257:     end;
258:     end;
259: end;
260: IStr[1]:='A';
261: IStr[2]:='B';
```

```

262:   IStr[3]:='C';
263:   IStr[4]:='D';
264:   IStr[5]:='E';
265: end;
266:
267: function GetRealFormat : String40;
268: var
269:   StrBuff, InsStr      : String40;
270:   I, J, SCode          : Integer;
271:   FoundIt              : Boolean;
272: begin
273:   StrBuff:=RealEq; I:=1; J:=1;
274:   repeat
275:     try
276:       if StrBuff[I]=IStr[J] then begin
277:         Str(Al[J]:5:2, InsStr);
278:         if Al[J]<0 then
279:           Delete(StrBuff, I-1, 2)
280:         else
281:           Delete(StrBuff, I, 1);
282:           Insert(InsStr, StrBuff, I);
283:           I:=I+Length(InsStr);
284:           Inc(J);
285:         end;
286:       finally
287:         Inc(I);
288:       end;
289:     until I>Length(StrBuff);
290:     GetRealFormat:=StrBuff;
291:   end;
292:
293: procedure Calculations;
294: var
295:   I      : Integer;
296: begin
297:   IAl[5]:=Sqrt(Sqr(Al[2])+Sqr(Al[3]+Al[5]*IAl[1]*IAl[4]))+
298:     Sqr(Al[4]+IAl[1]*IAl[3]));
299:   IAl[1]:=-1*(Al[1]/(Al[2]*IAl[2]+Al[3]*IAl[3]+Al[4]*IAl[4]+
300:     Al[5]*IAl[1]*IAl[3]*IAl[4]));
301:   IAl[2]:=-1*Al[2]/IAl[5];
302:   IAl[3]:=-1*(Al[3]+Al[5]*IAl[1]*IAl[4])/IAl[5];
303:   IAl[4]:=-1*(Al[4]+Al[5]*IAl[1]*IAl[3])/IAl[5];
304: end;
305:
306: procedure InputTable;
307: var
308:   I      : Integer;
309:   Takhir : Real;
310: begin
311:   NewTable.EmptyTable;
312:   I:=-1;
313:   repeat
314:     Inc(I);
315:     Takhir:=TAWal;
316:     Calculations;
317:     TAWal:=IAl[1];
318:     with NewTable do begin
319:       Active:=True;
320:       Append;
321:       Edit;
322:       if I=0 then
323:         begin
324:           FieldValues['Iterasi']:='Awal';
325:         try
326:           FieldValues['Beta']::=StrToFloat(AFOSM.IBeta.Text);
327:         except on EConvertError do

```



```
328:      begin
329:      MessageDlg('Nilai masukan "BETHA" harus berbentuk real',
330:                mtWarning, [mbOK], 0);
331:      AFOSM.IBeta.Text:='';
332:      AFOSM.Oke.Enabled:=False;
333:      end;
334:      end;
335:      try
336:      FieldValues['Alpha1']:=StrToFloat(AFOSM.IAlpha1.Text);
337:      except on EConvertError do
338:      begin
339:      MessageDlg('Nilai masukan "ALPHA 1" harus berbentuk real',
340:                mtWarning, [mbOK], 0);
341:      AFOSM.IAlpha1.Text:='';
342:      AFOSM.Oke.Enabled:=False;
343:      end;
344:      end;
345:      try
346:      FieldValues['Alpha2']:=StrToFloat(AFOSM.IAlpha2.Text);
347:      except on EConvertError do
348:      begin
349:      MessageDlg('Nilai masukan "ALPHA 2" harus berbentuk real',
350:                mtWarning, [mbOK], 0);
351:      AFOSM.IAlpha2.Text:='';
352:      AFOSM.Oke.Enabled:=False;
353:      end;
354:      end;
355:      try
356:      FieldValues['Alpha3']:=StrToFloat(AFOSM.IAlpha3.Text);
357:      except on EConvertError do
358:      begin
359:      MessageDlg('Nilai masukan "ALPHA 3" harus berbentuk real',
360:                mtWarning, [mbOK], 0);
361:      AFOSM.IAlpha3.Text:='';
362:      AFOSM.Oke.Enabled:=False;
363:      end;
364:      end;
365:      end
366:      else begin
367:      FieldValues['Iterasi']:=IntToStr(I);
368:      FieldValues['Beta']:=IAL[1];
369:      FieldValues['Alpha1']:=IAL[2];
370:      FieldValues['Alpha2']:=IAL[3];
371:      FieldValues['Alpha3']:=IAL[4];
372:      Post;
373:      end;
374:  end;
375:  with AFOSM.TempTable do begin
376:      Active:=True;
377:      Append;
378:      Edit;
379:      if I=0 then
380:      begin
381:      FieldValues['Iterasi']:='Awal';
382:      try
383:      FieldValues['Beta']:=StrToFloat(AFOSM.IBeta.Text);
384:      except on EConvertError do
385:      begin
386:      MessageDlg('Nilai masukan "BETHA" harus berbentuk real',
387:                mtWarning, [mbOK], 0);
388:      AFOSM.IBeta.Text:='';
389:      AFOSM.Oke.Enabled:=False;
390:      end;
391:      end;
392:      try
393:      FieldValues['Alpha1']:=StrToFloat(AFOSM.IAlpha1.Text);
```

```

394:         except on EConvertError do
395:         begin
396:             MessageDlg('Nilai masukan "ALPHA 1" harus berbentuk real',
397:                 mtWarning, [mbOK], 0);
398:             AFOSM.IAlpha1.Text:='';
399:             AFOSM.Oke.Enabled:=False;
400:         end;
401:     end;
402:     try
403:         FieldValues['Alpha2']:=StrToFloat(AFOSM.IAlpha2.Text);
404:         except on EConvertError do
405:         begin
406:             MessageDlg('Nilai masukan "ALPHA 2" harus berbentuk real',
407:                 mtWarning, [mbOK], 0);
408:             AFOSM.IAlpha2.Text:='';
409:             AFOSM.Oke.Enabled:=False;
410:         end;
411:     end;
412:     try
413:         FieldValues['Alpha3']:=StrToFloat(AFOSM.IAlpha3.Text);
414:         except on EConvertError do
415:         begin
416:             MessageDlg('Nilai masukan "ALPHA 3" harus berbentuk real',
417:                 mtWarning, [mbOK], 0);
418:             AFOSM.IAlpha3.Text:='';
419:             AFOSM.Oke.Enabled:=False;
420:         end;
421:     end;
422:     end
423:     else begin
424:         FieldValues['Iterasi']:=IntToStr(I);
425:         FieldValues['Beta']:=IAL[1];
426:         FieldValues['Alpha1']:=IAL[2];
427:         FieldValues['Alpha2']:=IAL[3];
428:         FieldValues['Alpha3']:=IAL[4];
429:         Post;
430:     end;
431: end;
432: until
433:     Abs(TAKhir-TAwal)<=0.0001;
434:     NewTable.Active:=False;
435:     AFOSM.TableOut.TableName:='c:\buffer.dbf';
436:     AFOSM.TableOut.Active:=True;
437: end;
438:
439: procedure InputGraphics;
440: begin
441:     AFOSM.GridOut.RefreshData;
442:     with AFOSM do begin
443:         Series1.DataSource:=TableOut;
444:         Series1.YValues.ValueSource:='Beta';
445:         Series2.DataSource:=TableOut;
446:         Series2.YValues.ValueSource:='Alpha1';
447:         Series3.DataSource:=TableOut;
448:         Series3.YValues.ValueSource:='Alpha2';
449:         Series4.DataSource:=TableOut;
450:         Series4.YValues.ValueSource:='Alpha3';
451:     end;
452:     AFOSM.PrintGraph.Refresh;
453:     with AFOSM do begin
454:         Series5.DataSource:=TableOut;
455:         Series5.YValues.ValueSource:='Beta';
456:         Series6.DataSource:=TableOut;
457:         Series6.YValues.ValueSource:='Alpha1';
458:         Series7.DataSource:=TableOut;
459:         Series7.YValues.ValueSource:='Alpha2';

```



```
460:     Series8.DataSource:=TableOut;
461:     Series8.YValues.ValueSource:='Alpha3';
462: end;
463: end;
464:
465: procedure TAFOSM.OkeClick(Sender: TObject);
466: begin
467:     if not FirstTime then begin
468:         CreateTableBuff;
469:         FirstTime:=True;
470:     end;
471:     AFOSM.TableOut.Active:=False;
472:     InputTable;
473:     InputGraphics;
474:     OutputAFOSM.Enabled:=True;
475:     PageControll1.Tag:=1;
476: end;
477:
478:
479: procedure TAFOSM.FormCreate(Sender: TObject);
480: begin
481:     OutputAFOSM.Enabled:=False;
482:     Initialize;
483:     Edit1.Text:=GetRealFormat;
484:     FirstTime:=False;
485:     NewTable:=TTable.Create(Self);
486:     with NewTable do
487:     begin
488:         Active:=False;
489:         DatabaseName := 'DBDEMOS';
490:         TableName := 'c:\buffer.dbf';
491:         TableType := ttDefault;
492:         FieldDefs.Clear;
493:         FieldDefs.Add('Iterasi', ftString, 4, False);
494:         FieldDefs.Add('Beta', ftFloat, 0, False);
495:         FieldDefs.Add('Alpha1', ftFloat, 0, False);
496:         FieldDefs.Add('Alpha2', ftFloat, 0, False);
497:         FieldDefs.Add('Alpha3', ftFloat, 0, False);
498:         CreateTable;
499:     end;
500:     with PageControll1 do
501:     begin
502:         if ActivePage=TabSheet3 then
503:         begin
504:             SpeedButton1.Enabled:=False;
505:             SpeedButton2.Enabled:=False;
506:             SpeedButton3.Enabled:=False;
507:         end;
508:
509:         if ActivePage=TabSheet2 then
510:         begin
511:             SpeedButton1.Enabled:=False;
512:             SpeedButton2.Enabled:=False;
513:             SpeedButton5.Enabled:=False;
514:             SpeedButton3.Enabled:=True;
515:         end;
516:
517:         if ActivePage=TabSheet1 then
518:         begin
519:             SpeedButton1.Enabled:=True;
520:             SpeedButton2.Enabled:=True;
521:             SpeedButton3.Enabled:=False;
522:         end;
523:     end;
524:
525: end;
```



```
526:
527: procedure TAFOSM.Exit1Click(Sender: TObject);
528: begin
529: Close;
530: end;
531:
532: procedure TAFOSM.PreviewClick(Sender: TObject);
533: begin
534: QuickRep1.Preview;
535: end;
536:
537: procedure TAFOSM.SpeedButton4Click(Sender: TObject);
538: begin
539: Close;
540: PilProg.Batal.Caption:='&Selesai';
541: PilProg.Advanced.checked:=false;
542: end;
543:
544: procedure TAFOSM.SpeedButton3Click(Sender: TObject);
545: begin
546: OutputAFOSM.Click;
547: end;
548:
549: procedure TAFOSM.SpeedButton2Click(Sender: TObject);
550: begin
551: New1.Click;
552: end;
553:
554:
555: procedure TAFOSM.New1Click(Sender: TObject);
556: begin
557: OutputAFOSM.Enabled:=False;
558: InputAFOSM.Enabled:=False;
559: SpeedButton1.Enabled:=False;
560: CoolBar1.Tag:=1;
561: PageControl1.Tag:=0;
562: IBeta.Text:='';
563: IAlpha1.Text:='';
564: IAlpha2.Text:='';
565: IAlpha3.Text:='';
566: Beta.Text:='';
567: Alpha1.Text:='';
568: Alpha2.Text:='';
569: Alpha3.Text:='';
570: Alpha4.Text:='';
571: Oke.Enabled:=False;
572:
573: end;
574:
575: procedure TAFOSM.IAlpha1Change(Sender: TObject);
576: begin
577: if
578: (IBeta.Text<>'') and
579: (IAlpha1.Text<>'') and
580: (IAlpha2.Text<>'') and
581: (Alpha3.Text<>'') and
582: (Beta.Text<>'') and
583: (Alpha1.Text<>'') and
584: (Alpha2.Text<>'') and
585: (Alpha3.Text<>'') and
586: (Alpha4.Text<>'')
587: then
588: Oke.Enabled:=True;
589: end;
590:
591: procedure TAFOSM.IAlpha2Change(Sender: TObject);
```

```
592: begin
593: if
594: (IBeta.Text<>'') and
595: (IAlpha1.Text<>'') and
596: (IAlpha2.Text<>'') and
597: (Alpha3.Text<>'') and
598: (Beta.Text<>'') and
599: (Alpha1.Text<>'') and
600: (Alpha2.Text<>'') and
601: (Alpha3.Text<>'') and
602: (Alpha4.Text<>'')
603: then
604: Oke.Enabled:=True;
605: end;
606:
607: procedure TAFOSM.IAlpha3Change(Sender: TObject);
608: begin
609: if
610: (IBeta.Text<>'') and
611: (IAlpha1.Text<>'') and
612: (IAlpha2.Text<>'') and
613: (Alpha3.Text<>'') and
614: (Beta.Text<>'') and
615: (Alpha1.Text<>'') and
616: (Alpha2.Text<>'') and
617: (Alpha3.Text<>'') and
618: (Alpha4.Text<>'')
619: then
620: Oke.Enabled:=True;
621: end;
622:
623: procedure TAFOSM.IBetaChange(Sender: TObject);
624: begin
625: if
626: (IBeta.Text<>'') and
627: (IAlpha1.Text<>'') and
628: (IAlpha2.Text<>'') and
629: (Alpha3.Text<>'') and
630: (Beta.Text<>'') and
631: (Alpha1.Text<>'') and
632: (Alpha2.Text<>'') and
633: (Alpha3.Text<>'') and
634: (Alpha4.Text<>'')
635: then
636: Oke.Enabled:=True;
637: end;
638:
639: procedure TAFOSM.BetaChange(Sender: TObject);
640: begin
641: if
642: (IBeta.Text<>'') and
643: (IAlpha1.Text<>'') and
644: (IAlpha2.Text<>'') and
645: (Alpha3.Text<>'') and
646: (Beta.Text<>'') and
647: (Alpha1.Text<>'') and
648: (Alpha2.Text<>'') and
649: (Alpha3.Text<>'') and
650: (Alpha4.Text<>'')
651: then
652: begin
653: Oke.Enabled:=True;
654: CoolBar1.Tag:=0;
655: end;
656: end;
657:
```

```
658: procedure TAFOSM.Alpha3Change(Sender: TObject);
659: begin
660:   if
661:     (IBeta.Text<>'') and
662:     (IAlpha1.Text<>'') and
663:     (IAlpha2.Text<>'') and
664:     (Alpha3.Text<>'') and
665:     (Beta.Text<>'') and
666:     (Alpha1.Text<>'') and
667:     (Alpha2.Text<>'') and
668:     (Alpha3.Text<>'') and
669:     (Alpha4.Text<>'')
670:   then
671:   begin
672:     Oke.Enabled:=True;
673:     CoolBar1.Tag:=0;
674:   end;
675: end;
676:
677: procedure TAFOSM.Alpha1Change(Sender: TObject);
678: begin
679:   if
680:     (IBeta.Text<>'') and
681:     (IAlpha1.Text<>'') and
682:     (IAlpha2.Text<>'') and
683:     (Alpha3.Text<>'') and
684:     (Beta.Text<>'') and
685:     (Alpha1.Text<>'') and
686:     (Alpha2.Text<>'') and
687:     (Alpha3.Text<>'') and
688:     (Alpha4.Text<>'')
689:   then
690:   begin
691:     Oke.Enabled:=True;
692:     CoolBar1.Tag:=0;
693:   end;
694: end;
695:
696: procedure TAFOSM.Alpha4Change(Sender: TObject);
697: begin
698:   if
699:     (IBeta.Text<>'') and
700:     (IAlpha1.Text<>'') and
701:     (IAlpha2.Text<>'') and
702:     (Alpha3.Text<>'') and
703:     (Beta.Text<>'') and
704:     (Alpha1.Text<>'') and
705:     (Alpha2.Text<>'') and
706:     (Alpha3.Text<>'') and
707:     (Alpha4.Text<>'')
708:   then
709:   begin
710:     Oke.Enabled:=True;
711:     CoolBar1.Tag:=0;
712:   end;
713: end;
714:
715: procedure TAFOSM.Alpha2Change(Sender: TObject);
716: begin
717:   if
718:     (IBeta.Text<>'') and
719:     (IAlpha1.Text<>'') and
720:     (IAlpha2.Text<>'') and
721:     (Alpha3.Text<>'') and
722:     (Beta.Text<>'') and
723:     (Alpha1.Text<>'') and
```



```
724: (Alpha2.Text<>'')and
725: (Alpha3.Text<>'')and
726: (Alpha4.Text<>'')
727: then
728: begin
729: Oke.Enabled:=True;
730: CoolBar1.Tag:=0;
731: end;
732: end;
733:
734: procedure TAFOSM.PageControl1Change(Sender: TObject);
735: begin
736: with PageControl1 do
737: begin
738: if ActivePage=TabSheet3 then
739: begin
740: SpeedButton1.Enabled:=False;
741: SpeedButton2.Enabled:=False;
742: SpeedButton3.Enabled:=False;
743: SpeedButton5.Enabled:=False;
744: end;
745:
746: if ActivePage=TabSheet2 then
747: begin
748: SpeedButton2.Enabled:=False;
749: SpeedButton1.Enabled:=False;
750: SpeedButton5.Enabled:=False;
751: if Tag=1 then
752: SpeedButton3.Enabled:=True;
753: end;
754:
755: if ActivePage=TabSheet1 then
756: begin
757: SpeedButton2.Enabled:=True;
758: SpeedButton3.Enabled:=False;
759: SpeedButton5.Enabled:=True;
760: if CoolBar1.Tag=0 then
761: SpeedButton1.Enabled:=True;
762: end;
763: end;
764: end;
765:
766: procedure TAFOSM.OutputAFOSMClick(Sender: TObject);
767: begin
768:   if SaveDialog1.Execute then begin
769:     SaveDialog1.FileName:='';
770:     UpDateTable:=TTable.Create(Self);
771:     with UpDateTable do
772:       begin
773:         Active:=False;
774:         DatabaseName := 'DBDEMOS';
775:         TableName := SaveDialog1.FileName;
776:         TableType := ttDefault;
777:         FieldDefs.Clear;
778:         FieldDefs.Add('Iterasi', ftString, 4, False);
779:         FieldDefs.Add('Beta', ftFloat, 0, False);
780:         FieldDefs.Add('Alpha1', ftFloat, 0, False);
781:         FieldDefs.Add('Alpha2', ftFloat, 0, False);
782:         FieldDefs.Add('Alpha3', ftFloat, 0, False);
783:         CreateTable;
784:       end;
785:       UpDateTable.BatchMove (AFOSM.TempTable,batAppend);
786:     end;
787:
788: end;
789:
```

```
790: procedure TAFOSM.SpeedButton1Click(Sender: TObject);
791: begin
792: InputAFOSM.Click;
793: end;
794:
795: procedure TAFOSM.InputAFOSMClick(Sender: TObject);
796: begin
797: SaveDialog2.FileName:='';
798: SaveDialog2.DefaultExt:='afs';
799: if SaveDialog2.Execute then
800: begin
801: Namafile:=SaveDialog2.FileName;
802: assignfile(InputPraktikum1,namafile);
803: rewrite(InputPraktikum1);
804: writeln(InputPraktikum1,Beta.Text);
805: writeln(InputPraktikum1,Alpha1.Text);
806: writeln(InputPraktikum1,Alpha2.Text);
807: writeln(InputPraktikum1,Alpha3.Text);
808: writeln(InputPraktikum1,Alpha4.Text);
809: writeln(InputPraktikum1,IAlpha1.Text);
810: writeln(InputPraktikum1,IAlpha2.Text);
811: writeln(InputPraktikum1,IAlpha3.Text);
812: writeln(InputPraktikum1,IBeta.Text);
813: Closefile(InputPraktikum1);
814: end;
815:
816: end;
817:
818: procedure TAFOSM.BukaFileInput1Click(Sender: TObject);
819: begin
820: OpenFileDialog1.FileName:='';
821: OpenFileDialog1.Filter := 'Text files (*.afs)|*.afs';
822: if OpenFileDialog1.Execute then
823: begin
824: NamaFile:=OpenDialog1.FileName;
825: try
826: AssignFile(InputPraktikum1,NamaFile);
827: reset(InputPraktikum1);
828: finally
829: with AFOSM do
830: begin
831: readln(InputPraktikum1,IsiData);
832: Beta.Text:=IsiData;
833: readln(InputPraktikum1,IsiData);
834: Alpha1.Text:=IsiData;
835: readln(InputPraktikum1,IsiData);
836: Alpha2.Text:=IsiData;
837: readln(InputPraktikum1,IsiData);
838: Alpha3.Text:=IsiData;
839: readln(InputPraktikum1,IsiData);
840: Alpha4.Text:=IsiData;
841: readln(InputPraktikum1,IsiData);
842: IAlpha1.Text:=IsiData;
843: readln(InputPraktikum1,IsiData);
844: IAlpha2.Text:=IsiData;
845: readln(InputPraktikum1,IsiData);
846: IAlpha3.Text:=IsiData;
847: readln(InputPraktikum1,IsiData);
848: IBeta.Text:=IsiData;
849: end;
850: CloseFile(InputPraktikum1);
851: AFOSM.Oke.Enabled:=True;
852: end;
853: end;
854:
855: end;
```

```
856:
857: procedure TAFOSM.SpeedButton5Click(Sender: TObject);
858: begin
859:   BukaFileInput1.Click;
860: end;
861:
862: end.
```



```

1: unit Hitung;
2:
3: interface
4: uses DBTables;
5: type
6:   A=1..10;
7:   B=1..3;
8:   C=1..50;
9:   D=1..51;
10:   Data=record
11:     NilaiRata:real;
12:     NilaiSBaku:real;
13:     Distribusi:boolean;
14:   end;
15:   F = array [A] of Data;
16:   G = array [A,D] of Real;
17:   H = array [A,C] of Real;
18:   M = array [B] of Real;
19:   TBigArray = array [1..1000] of Real;
20: var
21:   VarAcak: F;
22:   NTrial:LongInt;
23:   Nranvr,J:Byte;
24: procedure Utama;
25:
26: implementation
27: uses Unit3, Dialogs, SysUtils;
28: var
29:   Relsim: array [C] of Real;
30:   Tranvr: array [A,1..2,C] of real;
31:   IR : array [1..97] of integer;
32:   IntervalP,PDF:G;
33:   Trfmed: H;
34:   Rantrf: M;
35:   NFail,NProve,SafNeg,IFF:Integer;
36:   O:Word;
37:   L,P,Q: Byte;
38:   Nident,Idum,NCount,Experm,K:LongInt;
39:   Smrgin,Reliab,ProbFa,Rand,Sum,Clasin,BoundL,BoundU,Range:Real;
40:
41: Procedure Tabel;
42: (* DEVIDES THE RANGE OF A NORMAL RANDOM VARIABLE INTO 50 CLASS
43:   INTERVALS AND CALCULATES THE PROBILITY OF BEING IN THAT
44:   INTERVAL, AND RELATES THE CORESSPONDING VALUE*)
45: var
46:   VaRand:G;
47:   N,O,V:byte;
48:   W:LongInt;
49: begin
50:   Sum:=0;
51:   W:=0;
52:   for L :=1 to Nranvr do
53:     begin
54:       with VarAcak[L] do begin
55:         BoundL := NilaiRata - 4*NilaiSBaku;
56:         BoundU := NilaiRata + 4*NilaiSBaku;
57:         Range := BoundU - BoundL;
58:         Clasin := Range/50;
59:         if Distribusi = true then begin
60:           Unit3.SMC.ProgressBar1.Position:=0;
61:           for O := 1 to 51 do
62:             try
63:               if O=1 then PDF[L,1] := 0.0 else
64:                 PDF[L,O] := (1/(NilaiSBaku*sqrt(2*pi)))*exp(-0.5*
65:                 sqr(((BoundL+Clasin*(O-1))-NilaiRata)/NilaiSBaku));
66:             finally

```

```

67:         begin
68:         with NewTable[L] do
69:             begin
70:                 Active:=True;
71:                 Append;
72:                 Edit;
73:                 FieldValues['Interval']:=0;
74:                 FieldValues['FKP']:= PDF[L,O];
75:                 Post;
76:             end;
77:         end;
78:         with SMC.Table1 do
79:             begin
80:                 Active:=True;
81:                 Append;
82:                 Edit;
83:                 FieldValues['Interval']:=0;
84:                 FieldValues['FKP']:= PDF[L,O];
85:                 Post;
86:             end;
87:         end;
88:     end {Akhir Kondisi Distribusi Normal}
89:     else begin {Awal Kondisi Log Normal}
90:         for P:= 1 to 51 do begin
91:             try
92:                 begin
93:                     PDF[L,P]:=1/((BoundL+Clasin*(P-1))*NilaiSBaku*
94:                     sqrt(2*pi))*exp(-0.5*
95:                     sqr((BoundL+Clasin*(P-1))-NilaiRata)/NilaiSBaku));
96:                     VaRand[L,1]:=BoundL;
97:                     for O:=2 to 51 do
98:                         VaRand[L,O]:=VaRand[L,O-1]+Clasin;
99:                     end;
100:                 finally
101:                     with NewTable[L] do
102:                         begin
103:                             Active:=True;
104:                             Append;
105:                             Edit;
106:                             FieldValues['Interval']:=0;
107:                             FieldValues['FKP']:= PDF[L,P];
108:                             Post;
109:                         end;
110:                     with SMC.Table1 do
111:                         begin
112:                             Active:=True;
113:                             Append;
114:                             Edit;
115:                             FieldValues['Interval']:=0;
116:                             FieldValues['FKP']:= PDF[L,P];
117:                             Post;
118:                         end;
119:                     end; {Akhir Try}
120:                 end; {Looping P}
121:             end; {Kondisi Distribusi Log Normal}
122:             NewTable[L].Active:=False;
123:             SMC.Table2.Active:=false;
124:             SMC.Table2.TableName:='c:\bufer1.dbf';
125:             SMC.Table2.Active:=true;
126:             for V:=1 to 51 do
127:                 begin
128:                     end;
129:
130:                 end; {Akhir Looping VarAcak}
131:                 SMC.Prev.enabled:=false;
132:                 for N:=1 to 50 do

```

```

133:         begin
134:             Tranvr[L,1,N]:=(PDF[L,N]+PDF[L,N+1])/2;
135:             Trfmed[L,N]:=Varand[L,N]+Varand[L,N+1]/2;
136:             Tranvr[L,2,N]:=Tranvr[L,1,N]*Clasin;
137:             Sum:=Sum+Tranvr[L,2,N];
138:         end; {Akhir Looping N}
139:     for P:= 1 to 51 do
140:         IntervalP[L,P]:=0.0;
141:         IntervalP[L,1]:=0.0;
142:         IntervalP[L,51]:=1.0;
143:         for Q:= 2 to 51 do
144:             begin
145:                 if (Q >= 2) and (Q<= 50) then
146:                     IntervalP[L,Q]:=IntervalP[L,Q-1]+Tranvr[L,2,Q];
147:                 end; {Akhir Looping Q}
148:             end; {Akhir Looping L}
149:         end;
150:
151: procedure Input;
152: (* *)
153: var
154:     ed,di:Word;
155:     de,id:Byte;
156: begin
157:     ed:=NTrial;
158:     de:=NRanvr;
159:     di:=ed;
160:     id:=de;
161:     Tabel;
162:     Ntrial:=di;
163:     Nranvr:=id;
164: end;
165:
166:
167: procedure Random{(var Rand: real)};
168: (* *)
169: var
170:     IY:LongInt;
171:     RM,S:Double;
172:     U:Byte;
173: const
174:     R=714025;
175:     IA=1366;
176:     IC=150889;
177: begin
178:     IFF:=0;
179:     RM:=1/R;
180:     if (Idum<0) or (IFF=0) then
181:     begin
182:         IFF:=1;
183:         Idum:=((IC-IDUM) mod R);
184:         for U:= 1 to 97 do
185:             begin
186:                 Idum:=((IA*Idum+IC) mod R);
187:                 IR[U]:= Idum;
188:             end;
189:             Idum:= ((IA*Idum+IC) mod R);
190:             IY := Idum;
191:         end;
192:         U:=1+ abs((97*IY) div R);
193:         try
194:             IY:=IR[U];
195:             Rand:=IY*RM;
196:             Idum:=((IA*Idum+IC) mod R);
197:             IR[U]:= Idum;

```



```
199:         except on ERangeError do MessageDlg('Nilai U = '+IntToStr(U),
200:                                         MtInformation, [mbOK], 0);
201:     end;
202: end;
203:
204:
205: Procedure Transformasi;
206: (* *)
207: var
208:     Ran: real;
209: begin
210:     for L:=1 to Nranvr do
211:     begin
212:         NCount:=NCount+1;
213:         Rantrf[L]:=0.0;
214:         Random;
215:         Ran:=Rand;
216:         for O:= 1 to 50 do
217:         begin
218:             if (IntervalP[L,O]<Ran) and (Ran<IntervalP[L,O+1]) then
219:                 Rantrf[L]:= Trfmed[L,O];
220:             end;
221:         end;
222:     end;
223:
224: Procedure SafetyMargin;
225: (* *)
226: begin
227:     try
228:         Smrgin:= 1-(2.0E-05*Rantrf[1])-(2.0E-04*Rantrf[2])-(
229:                 (2.0E-5*Rantrf[3]));
230:     finally
231:     end;
232: end;
233:
234:
235: Procedure Output;
236: (* *)
237: var
238:     J: Integer;
239:
240: BEGIN
241:     SMC.Label12.Caption:=IntToStr(Experm);
242:     SMC.Label13.Caption:=IntToStr(NRanvr);
243:     SMC.Label14.Caption:=FloatToStr(Probfa);
244:     SMC.Label15.Caption:=FloatToStr(Reliab);
245:
246:     SMC.DBChart1.Refresh;
247:     With SMC Do Begin
248:         Series1.DataSource:=SMC.Table2;
249:         Series1.YValues.ValueSource:='FKP';
250:     End;
251:
252: END;
253:
254: (*MAIN PROGRAM*)
255: procedure Utama;
256: var
257:     T:Double;
258:     U:String[4];
259:
260: begin
261:     Idum:=-100;
262:     Nfail:=0;
263:     Ncount:=0;
264:     Nprove:=0;
```

```
265:     U:='';
266:     Nident:=Ntrial*Nranvr;
267:     Tabel;
268:     SMC.ProgressBar1.Max:=Ntrial;
269:     repeat
270:         Nprove:=Nprove+1;
271:         SMC.ProgressBar1.Position:=0;
272:         for K:= 1 to Ntrial do
273:             begin
274:                 SMC.ProgressBar1.Position:=SMC.ProgressBar1.Position+1;
275:                 Transformasi;
276:                 if NIdent-NCount=50 then
277:                     begin
278:                         MessageDlg('Simulasi Selesai',mtInformation,[mbOK],0);
279:                     end;
280:                 SafetyMargin;
281:                 if Smrgin<0.0 then
282:                     NFail:=Nfail+1;
283:                 end;
284:                 Safneg:=Nfail;
285:                 Experm:=Nprove*Ntrial*Nranvr;
286:                 PROBFA:=Safneg/Experm;
287:                 Reliab:=1-Probfa;
288:                 try
289:                     Relsim[Nprove]:=Reliab;
290:                     except on ERangeError do
291:                         Nprove:=Nprove-1;
292:                     end;
293:                 if Nprove=1 then
294:                     T:=Relsim[Nprove] else
295:                     begin
296:                         try
297:                             T:=abs(Relsim[Nprove]-Relsim[Nprove-1]);
298:                         finally
299:                             end;
300:                         end;
301:                 if Nprove=10 then
302:                     begin
303:                         T:=0.00001;
304:                         U:='Stop';
305:                         MessageDlg('Proses dihentikan, karena jumlah
306:                             perhitungan terlalu kecil...',mtError,[mbOK],0);
307:                     end;
308:                 until (T<=0.0001);
309:                 if U<>'Stop' then
310:                     begin
311:                         Output;
312:                     end;
313:             end;
314:         end.
```

```
1: unit First;
2:
3: interface
4:
5: uses
6:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
7:   Dialogs, ToolWin, ComCtrls, Menus;
8:
9: type
10:   TForm1 = class(TForm)
11:     MainMenu1: TMainMenu;
12:     File1: TMenuItem;
13:     New1: TMenuItem;
14:     Save1: TMenuItem;
15:     Print1: TMenuItem;
16:     Exit1: TMenuItem;
17:     ImageList1: TImageList;
18:     Add1: TMenuItem;
19:     TopEvent1: TMenuItem;
20:     IntermediateEvent1: TMenuItem;
21:     PrimeEvent1: TMenuItem;
22:     ORGate1: TMenuItem;
23:     ANDGate1: TMenuItem;
24:     Help1: TMenuItem;
25:     About1: TMenuItem;
26:     Help2: TMenuItem;
27:     procedure Exit1Click(Sender: TObject);
28:   private
29:     { Private declarations }
30:   public
31:     { Public declarations }
32:   end;
33:
34: var
35:   Form1: TForm1;
36:
37: implementation
38:
39: {$R *.DFM}
40:
41:
42:
43: procedure TForm1.Exit1Click(Sender: TObject);
44: begin
45:   Close;
46: end;
47:
48:
49: end.
```



```
1: unit GateForm;
2:
3: interface
4:
5: uses
6:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
7:   Dialogs, ExtCtrls, Buttons, StdCtrls;
8:
9: type
10:   TGForm = class(TForm)
11:     RadioButton1: TRadioButton;
12:     RadioButton2: TRadioButton;
13:     SpeedButton1: TSpeedButton;
14:     procedure FormActivate(Sender: TObject);
15:     procedure SpeedButton1Click(Sender: TObject);
16:   private
17:     function Execute: Boolean;
18:     { Private declarations }
19:   public
20:     function GetRadioStat(var Stat : string): Boolean;
21:     { Public declarations }
22:   end;
23:
24: var
25:   GForm: TGForm;
26:
27: implementation
28:
29:   {$R *.DFM}
30:
31: procedure TGForm.FormActivate(Sender: TObject);
32: begin
33:   RadioButton1.Checked:=True;
34: end;
35:
36: function TGForm.Execute: Boolean;
37: begin
38:   Result := ShowModal = mrOk;
39: end;
40:
41: function TGForm.GetRadioStat(var Stat: string): Boolean;
42: Begin
43:   If RadioButton1.Checked Then Stat:='AND'
44:   Else Stat:='OR'
45: End;
46:
47: procedure TGForm.SpeedButton1Click(Sender: TObject);
48: begin
49:   GForm.Close;
50: end;
51:
52: end.
```

```
1: unit proper;
2:
3: interface
4:
5: uses
6:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dial
7:   StdCtrls, ExtCtrls, Buttons;
8:
9: type
10:   TPropForm = class(TForm)
11:     Label1: TLabel;
12:     Label2: TLabel;
13:     Edit1: TEdit;
14:     Edit2: TEdit;
15:     SpeedButton1: TSpeedButton;
16:     procedure SpeedButton1Click(Sender: TObject);
17:   private
18:     { Private declarations }
19:   public
20:     { Public declarations }
21:   end;
22:
23: var
24:   PropForm: TPropForm;
25:
26: implementation
27:
28: {$R *.DFM}
29:
30:
31:
32: procedure TPropForm.SpeedButton1Click(Sender: TObject);
33: begin
34:   PropForm.Close;
35: end;
36:
37: end.
```



```
1: unit Unit5;
2:
3: interface
4:
5: uses
6:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
7:   Dialogs, Menus, FTAEvent, FTAGate, ExtCtrls, Tabs, StdCtrls, Db,
8:   DBTables;
9:
10: type
11:   TForm2 = class(TForm)
12:     MainMenu1: TMainMenu;
13:     File1: TMenuItem;
14:     New1: TMenuItem;
15:     Open1: TMenuItem;
16:     Save1: TMenuItem;
17:     Exit1: TMenuItem;
18:     Edit1: TMenuItem;
19:     TopEvent1: TMenuItem;
20:     IntermediateEvent1: TMenuItem;
21:     Primeevent1: TMenuItem;
22:     ORGate1: TMenuItem;
23:     ANDGate1: TMenuItem;
24:     Edit2: TMenuItem;
25:     Cut1: TMenuItem;
26:     Help1: TMenuItem;
27:     About1: TMenuItem;
28:     OpenDialog1: TOpenDialog;
29:     SaveDialog1: TSaveDialog;
30:     PopupMenu1: TPopupMenu;
31:     Cut2: TMenuItem;
32:     Properties1: TMenuItem;
33:     ScrollBox1: TScrollBox;
34:     Panel1: TPanel;
35:     Label1: TLabel;
36:     Label2: TLabel;
37:     Label3: TLabel;
38:     Label4: TLabel;
39:     TabSet1: TTabSet;
40:     Notebook1: TNotebook;
41:     PaintBox1: TPaintBox;
42:     Properties2: TMenuItem;
43:     TableSave: TTable;
44:     DataSource1: TDataSource;
45:     procedure Exit1Click(Sender: TObject);
46:     procedure RedrawDown(Noevent, BesarGeser: integer);
47:     procedure CheckArea(IGeser: integer; ClickX, ClickY: integer);
48:     procedure Save1Click(Sender: TObject);
49:     procedure Open1Click(Sender: TObject);
50:     procedure FormClose(Sender: TObject; var Action: TCloseAction);
51:     procedure FormCreate(Sender: TObject);
52:     procedure ParamEvent(NoGateUp: integer; TextEvent: String);
53:     procedure ParameterGate(PosX, PosY, NoEventUp: integer;
54:       TextGate: String);
55:     procedure PaintBox1MouseDown(Sender: TObject; Button:
56:       TMouseButton; Shift: TShiftState; X, Y: Integer);
57:     procedure TengahinEvent;
58:     procedure SetCursor;
59:     procedure DelDownEvent(EventTop: integer);
60:     procedure DelDownGate(GateTop: integer);
61:     procedure DelEvent(DarkI: integer);
62:     procedure DelGate(DarkGate: integer);
63:     procedure FormDestroy(Sender: TObject);
64:     procedure FormActivate(Sender: TObject);
65:     procedure TopEvent1Click(Sender: TObject);
66:     procedure IntermediateEvent1Click(Sender: TObject);
67:     procedure Primeevent1Click(Sender: TObject);
68:     procedure ORGate1Click(Sender: TObject);
69:     procedure ANDGate1Click(Sender: TObject);
70:     procedure ScrollToDark;
```



```

71:     procedure Delete2Click(Sender: TObject);
72:     procedure PaintBox1Paint(Sender: TObject);
73:     procedure Properties2Click(Sender: TObject);
74:     procedure Properties1Click(Sender: TObject);
75:     procedure Cut1Click(Sender: TObject);
76:     procedure Cut2Click(Sender: TObject);
77:     procedure PaintBox1Db1Click(Sender: TObject);
78:     procedure BikinMDIChild(const Name: string);
79:     procedure New1Click(Sender: TObject);
80:   private
81:     { Private declarations }
82:     DataEvent: TListData;
83:     DataGate: TListGate;
84:     I: Integer;
85:     Igate: Integer;
86:     DarkI: integer;
87:     DarkGate: integer;
88:     PEtext: String;
89:     IEtext: String;
90:     TEvent, IEvent, PEvent, AButton, OButton : Boolean;
91:   public
92:     { Public declarations }
93:     Drawing: Boolean;
94:   end;
95:
96: var
97:   Form2: TForm2;
98:
99: implementation
100:
101: {$R *.DFM}
102:
103: uses First, Unit3, Unit4, DataRec, DRecGate, proper, GateForm;
104:
105: Procedure ReCalculate;
106: Var
107:   ILoop, INGate, INLoop      : Integer;
108:   TotalProb, OProb          : Real;
109:   EndFound, EField : Boolean;
110: Begin
111:   With Form2 Do Begin
112:     ILoop:=DataGate.Count;
113:     EndFound:=False;
114:     Repeat
115:       EField:=False;
116:       INLoop:=DataEvent.Count;
117:       TotalProb:=1;
118:       Repeat
119:         If DataEvent.Wilayah[INLoop].NumberGateUp=ILoop
120:         Then
121:           If DataGate.WilGate[ILoop].Textnya='AND'
122:           Then
123:             TotalProb:=TotalProb*DataEvent.Wilayah[INLoop].Probability
124:           Else
125:             TotalProb:=TotalProb+DataEvent.Wilayah[INLoop].Probability;
126:           Dec(INLoop);
127:         Until (INLoop<=0);
128:         If DataGate.WilGate[ILoop].Textnya='OR'
129:         Then TotalProb:=TotalProb-1;
130:         DataEvent.Wilayah[DataGate.WilGate[ILoop].NumberEventUp].
131:           Probability:=TotalProb;
132:         Dec(ILoop);
133:       Until ILoop<=0;
134:     End;
135:   End;
136:
137:
138: procedure TForm2.RedrawDown(Noevent, BesarGeser: integer);
139: var
140:   NoGate, IRed: integer;

```

```

141: Begin
142:   DataEvent.Wilayah[Noevent].KoordX:=DataEvent.Wilayah[Noevent].
143:   KoordX+BesarGeser;
144:   if (DataEvent.Wilayah[Noevent].NumberGateUp<>0) and (DataGate.
145:   WilGate[DataEvent.Wilayah[Noevent].NumberGateUp].
146:   NumberOfEvent>1)
147:   then
148:     DataEvent.Wilayah[Noevent].KoordY:=DataGate.WilGate[DataEvent.
149:     Wilayah[Noevent].NumberGateUp].KoordY+30+DataEvent.
150:     Wilayah[Noevent].Tinggi div 2
151:   else if DataEvent.Wilayah[Noevent].NumberGateUp<>0
152:   then
153:     DataEvent.Wilayah[Noevent].KoordY:=DataGate.WilGate[DataEvent.
154:     Wilayah[Noevent].NumberGateUp].KoordY+15+DataEvent.
155:     Wilayah[Noevent].Tinggi div 2;
156:   NoGate:=DataEvent.Wilayah[Noevent].NumberGateDown;
157:   if (NoGate<>0) then
158:     Begin
159:       DataGate.WilGate[Nogate].KoordX:=DataGate.WilGate[Nogate].
160:       KoordX+BesarGeser;
161:       DataGate.WilGate[Nogate].KoordY:=(DataEvent.Wilayah[Noevent].
162:       KoordY+65+DataEvent.Wilayah[Noevent].Tinggi div 2)-(DataEvent.
163:       Wilayah[Noevent].KoordY+15+DataEvent.Wilayah[Noevent].Tinggi
164:       div 2))div 2+(DataEvent.Wilayah[Noevent].KoordY+15+DataEvent.
165:       Wilayah[Noevent].Tinggi div 2);
166:       If (DataGate.WilGate[Nogate].NumberOfEvent>1)
167:       then
168:         Begin
169:           IRed:=0;
170:           repeat
171:             inc(IRed);
172:             if (DataEvent.Wilayah[IRed].NumberGateUp=NoGate) then
173:               RedrawDown(IRed,BesarGeser);
174:             Until IRed=DataEvent.Count;
175:           end
176:         else
177:           Begin
178:             IRed:=0;
179:             repeat
180:               inc(IRed);
181:               if (DataEvent.Wilayah[IRed].NumberGateUp=Nogate) then
182:                 RedrawDown(IRed,BesarGeser);
183:               Until IRed=DataEvent.Count;
184:             end;
185:           end;
186:         end;
187:
188: procedure TForm2.CheckArea(IGeser: integer; ClickX,ClickY: integer);
189: var
190:   LoopI,LLoopI,ILoopI,BesarGeserL,BesarGeserR,BesarGeserC: integer;
191:   LoopIT,LoopIB,LoopIL,LoopIR: integer;
192:   IIncT,IIncB,IIncL,IIncR: integer;
193:   BL,BR,Check,Diantara: boolean;
194:   IBL,IBR: integer;
195:   NoEvent,NoGate,NomorEvent,NomorGate: integer;
196: Begin
197:   IIncT:=ClickY-DataEvent.Wilayah[IGeser].Tinggi div 2;
198:   IIncB:=ClickY+DataEvent.Wilayah[IGeser].Tinggi div 2;
199:   IIncL:=ClickX-DataEvent.Wilayah[IGeser].Lebar div 2;
200:   IIncR:=ClickX+DataEvent.Wilayah[IGeser].Lebar div 2;
201:   LoopI:=0;
202:   ILoopI:=0;
203:   BL:=False;
204:   IBL:=0;
205:   BR:=False;
206:   IBR:=0;
207:   Diantara:=False;
208:   repeat
209:     inc(LoopI);
210:     LoopIT:=DataEvent.Wilayah[LoopI].koordY-DataEvent.

```



```

211:           Wilayah[LoopI].Tinggi div 2;
212:   LoopIB:=DataEvent.Wilayah[LoopI].koordY+DataEvent.
213:           Wilayah[LoopI].Tinggi div 2;
214:   LoopIL:=DataEvent.Wilayah[LoopI].koordX-DataEvent.
215:           Wilayah[LoopI].Lebar div 2;
216:   LoopIR:=DataEvent.Wilayah[LoopI].koordX+DataEvent.
217:           Wilayah[LoopI].Lebar div 2;
218:   Check:=False;
219:   if (IIncT>=LoopIT) and (IIncT<=LoopIB) then
220:     Check:=True
221:   else if (IIncB>=LoopIT) and (IIncB<=LoopIB) then
222:     Check:=True;
223:   if (Check=True) then
224:   Begin
225:     if (IIncL>=LoopIL) and (IIncL<=LoopIR) and (LoopI<>IGeser) then
226:     Begin
227:       IBL:=LoopI;
228:       NoEvent:=IBL;
229:       NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
230:       NomorEvent:=IGeser;
231:       NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
232:       while NoGate<>NomorGate do
233:       Begin
234:         NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
235:         NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
236:         NomorEvent:=DataGate.WilGate[NomorGate].NumberEventUp;
237:         NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
238:       end;
239:       if (NoEvent<NomorEvent) then
240:         BL:=True
241:       else if (NoEvent>NomorEvent) then
242:       Begin
243:         BR:=True;
244:         IBR:=IBL;
245:       end;
246:     end
247:   else if (IIncR>=LoopIL) and (IIncR<=LoopIR) and (LoopI<>IGeser)
248:   then Begin
249:     IBR:=LoopI;
250:     NoEvent:=IBR;
251:     NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
252:     NomorEvent:=IGeser;
253:     NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
254:     while NoGate<>NomorGate do
255:     Begin
256:       NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
257:       NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
258:       NomorEvent:=DataGate.WilGate[NomorGate].NumberEventUp;
259:       NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
260:     end;
261:     if (NoEvent<NomorEvent) then
262:     Begin
263:       BL:=True;
264:       IBL:=IBR;
265:     end
266:   else if (NoEvent>NomorEvent) then
267:     BR:=True;
268:   end
269:   else if (DataEvent.Wilayah[IGeser].NumberGateUp<>DataEvent.
270:   Wilayah[LoopI].NumberGateUp)
271:   then Begin
272:     LLoopI:=0;
273:     repeat
274:     inc(LLoopI);
275:     if (DataEvent.Wilayah[LoopI].NumberGateUp=DataEvent.
276:     Wilayah[LLoopI].NumberGateUp) and (LoopI<>LLoopI) and
277:     (DataEvent.Wilayah[LoopI].KoordX<DataEvent.
278:     Wilayah[IGeser].KoordX) and (DataEvent.Wilayah[LLoopI].
279:     KoordX>DataEvent.Wilayah[IGeser].KoordX) then
280:     Begin

```



```

281:         Diantara:=true;
282:         ILoopI:=LoopI;
283:     end;
284:     Until LLoopI=DataEvent.Count;
285: end;
286: end;
287: Until LoopI=DataEvent.Count;
288: if (BL=True) and (BR=True) and (IBL<>IBR) then
289: Begin
290:     BesarGesarL:=- (DataEvent.Wilayah[IBL].KoordX+DataEvent.
291:     Wilayah[IBL].Lebar div 2-IIIncL+10);
292:     NoEvent:=IBL;
293:     NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
294:     NomorEvent:=IGeser;
295:     NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
296:     while NoGate<>NomorGate do
297:     Begin
298:         NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
299:         NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
300:         NomorEvent:=DataGate.WilGate[NomorGate].NumberEventUp;
301:         NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
302:     end;
303:     LoopI:=0;
304:     repeat
305:         inc(LoopI);
306:         if (DataEvent.Wilayah[LoopI].NumberGateUp=DataEvent.
307:         Wilayah[NoEvent].NumberGateUp) and (LoopI<NoEvent)
308:         then
309:             RedrawDown (LoopI, BesarGesarL);
310:         Until LoopI=DataEvent.Count;
311:         RedrawDown (NoEvent, BesarGesarL);
312:         BesarGesarR:=- (DataEvent.Wilayah[IBR].KoordX-DataEvent.
313:             Wilayah[IBR].Lebar
314:             div 2)+10;
315:         NoEvent:=IBR;
316:         NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
317:         NomorEvent:=IGeser;
318:         NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
319:         while NoGate<>NomorGate do
320:         Begin
321:             NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
322:             NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
323:             NomorEvent:=DataGate.WilGate[NomorGate].NumberEventUp;
324:             NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
325:         end;
326:         LoopI:=0;
327:         repeat
328:             inc(LoopI);
329:             if (DataEvent.Wilayah[LoopI].NumberGateUp=DataEvent.
330:             Wilayah[NoEvent].NumberGateUp) and (LoopI>NoEvent)
331:             then
332:                 RedrawDown (LoopI, BesarGesarR);
333:             Until LoopI=DataEvent.Count;
334:             RedrawDown (NoEvent, BesarGesarR);
335:         end
336:     else if (BL=True) then
337:     Begin
338:         BesarGesarL:=- (DataEvent.Wilayah[IBL].KoordX+DataEvent.
339:         Wilayah[IBL].Lebar div 2-IIIncL+10);
340:         NoEvent:=IBL;
341:         NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
342:         NomorEvent:=IGeser;
343:         NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
344:         while NoGate<>NomorGate do
345:         Begin
346:             NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
347:             NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
348:             NomorEvent:=DataGate.WilGate[NomorGate].NumberEventUp;
349:             NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
350:         end;

```

```

351:   LoopI:=0;
352:   repeat
353:     inc(LoopI);
354:     if (DataEvent.Wilayah[LoopI].NumberGateUp=DataEvent.
355:       Wilayah[NoEvent].NumberGateUp) and (LoopI<NoEvent)
356:     then
357:       RedrawDown (LoopI, BesarGeserL);
358:   Until LoopI=DataEvent.Count;
359:   RedrawDown (NoEvent, BesarGeserL);
360: end
361: else if (BR=True) then
362: Begin
363:   LoopI:=0;
364:   repeat
365:     inc(LoopI);
366:     if (DataEvent.Wilayah[LoopI].NumberGateUp=DataEvent.
367:       Wilayah[IBR].NumberGateUp) and (LoopI<IBR)
368:     then
369:       IBR:=LoopI;
370:   Until LoopI=DataEvent.Count;
371:   BesarGeserR:=IIncR- (DataEvent.Wilayah[IBR].KoordX-DataEvent.
372:     Wilayah[IBR].Lebar div 2)+10;
373:   NoEvent:=IBR;
374:   NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
375:   NomorEvent:=IGeser;
376:   NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
377:   while NoGate<>NomorGate do
378:   Begin
379:     NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
380:     NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
381:     NomorEvent:=DataGate.WilGate[NomorGate].NumberEventUp;
382:     NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
383:   end;
384:   LoopI:=0;
385:   repeat
386:     inc(LoopI);
387:     if (DataEvent.Wilayah[LoopI].NumberGateUp=DataEvent.
388:       Wilayah[NoEvent].NumberGateUp) and (LoopI>NoEvent)
389:     then
390:       RedrawDown (LoopI, BesarGeserR);
391:   Until LoopI=DataEvent.Count;
392:   RedrawDown (NoEvent, BesarGeserR);
393: end
394: else if (Diantara=True) then
395: Begin
396:   NoEvent:=ILoopI;
397:   NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
398:   NomorEvent:=IGeser;
399:   NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
400:   while NoGate<>NomorGate do
401:   Begin
402:     NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
403:     NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
404:     NomorEvent:=DataGate.WilGate[NomorGate].NumberEventUp;
405:     NomorGate:=DataEvent.Wilayah[NomorEvent].NumberGateUp;
406:   end;
407:   if NomorEvent>NoEvent then
408:   Begin
409:     LoopI:=0;
410:     BesarGeserC:=0;
411:     repeat
412:       inc(LoopI);
413:       if (LoopI>ILoopI) and (DataEvent.Wilayah[LoopI].NumberGateUp=
414:         DataEvent.Wilayah[ILoopI].NumberGateUp)
415:       then
416:         BesarGeserC:=DataEvent.Wilayah[LoopI].KoordX-DataEvent.
417:           Wilayah[IGeser].KoordX+DataEvent.Wilayah[IGeser].Lebar+10;
418:       until LoopI=DataEvent.Count;
419:     LoopI:=0;
420:     repeat

```



```

421:         inc(LoopI);
422:         if (LoopI>NomorEvent) and (DataEvent.Wilayah[LoopI].
423:         NumberGateUp=DataEvent.Wilayah[NomorEvent].NumberGateUp)
424:         then
425:             RedrawDown(LoopI, BesarGesarC);
426:         Until LoopI=DataEvent.Count;
427:         RedrawDown(NomorEvent, BesarGesarC);
428:     end
429: else
430:     Begin
431:         BesarGesarC:=- (DataEvent.Wilayah[IGesar].KoordX-DataEvent.
432:         Wilayah[ILoopI].KoordX)-DataEvent.Wilayah[IGesar].Lebar-10;
433:         LoopI:=0;
434:         repeat
435:             inc(LoopI);
436:             if (LoopI<NomorEvent) and (DataEvent.Wilayah[LoopI].
437:             NumberGateUp=DataEvent.Wilayah[NomorEvent].NumberGateUp)
438:             then
439:                 RedrawDown(LoopI, BesarGesarC);
440:             Until LoopI=DataEvent.Count;
441:             RedrawDown(NomorEvent, BesarGesarC);
442:         end;
443:     end;
444: end;
445:
446: procedure TForm2.Exit1Click(Sender: TObject);
447: begin
448:     Close;
449: end;
450:
451: procedure TForm2.Save1Click(Sender: TObject);
452: Var
453:     MyTable, OutTable : TTable;
454:     ISave : Integer;
455: begin
456:     SaveDialog1.Execute;
457:     If SaveDialog1.FileName<>' ' Then
458:     Begin
459:         MyTable:=TTable.Create(Self);
460:         With MyTable Do Begin
461:             Active:=False;
462:             TableName := SaveDialog1.FileName;
463:             FieldDefs.Clear;
464:             FieldDefs.Add('Number', ftInteger, 0, False);
465:             FieldDefs.Add('Event', ftString, 20, False);
466:             FieldDefs.Add('Prob', ftFloat, 0, False);
467:             CreateTable;
468:             ISave:=0;
469:             Active:=True;
470:             Repeat
471:                 Inc(ISave);
472:                 Append;
473:                 Edit;
474:                 FieldValues['Number']:=ISave;
475:                 FieldValues['Event']:=DataEvent.Wilayah[ISave].Textnya;
476:                 FieldValues['Prob']:=DataEvent.Wilayah[ISave].Probability;
477:                 Post;
478:             Until ISave>=DataEvent.Count;
479:             End;
480:             MyTable.Active:=False;
481:         End;
482:     end;
483: end;
484:
485: procedure TForm2.Open1Click(Sender: TObject);
486: begin
487:     OpenDialog1.Execute;
488: end;
489:
490: procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);

```



```

491: begin
492:   Action:=caFree;
493:   TEvent:=False;
494:   IEvent:=False;
495:   PEvent:=False;
496:   AButton:=False;
497:   OButton:=False;
498: end;
499:
500: procedure TForm2.FormCreate(Sender: TObject);
501: var
502:   x:integer;
503:   y:integer;
504: begin
505:   DataEvent:=TListData.Create(1);
506:   DataGate:=TListGate.Create(1);
507:   Tabset1.Tabs:=Notebook1.Pages;
508:   Notebook1.PageIndex:=TabSet1.TabIndex;
509:   I:=0;
510:   Igate:=0;
511:   Drawing:=True;
512:   X:=320;
513:   Y:=50;
514:   ParamEvent(0, 'Top Event');
515:   ParameterGate(X,Y,1, 'AND');
516:   DarkI:=0;
517:   DarkGate:=0;
518:   PaintBox1.Refresh;
519: end;
520:
521: procedure TForm2.ParamEvent(NoGateUp: integer;TextEvent: String);
522: var
523:   Ad: PDataRec;
524:   DataRec: TDataRec;
525:   IInc: integer;
526: Begin
527:   i:=i+1;
528:   if i>DataEvent.Count then
529:     Begin
530:       DataRec:=TDataRec.Create;
531:       DataEvent.Expand;
532:       New(Ad);
533:       Ad^:=DataRec;
534:       DataEvent.Add(Ad);
535:     end;
536:   DataEvent.Wilayah[i].TypeEvent:=TextEvent;
537:   DataEvent.Wilayah[i].Textnya:=TextEvent;
538:   DataEvent.Wilayah[i].NumberGateUp:=NoGateUp;
539:   DataEvent.Wilayah[i].Lebar:=100;
540:   DataEvent.Wilayah[i].Tinggi:=50;
541:   if NoGateUp<>0 then
542:     Begin
543:       If DataGate.WilGate[NoGateUp].NumberOfEvent<>0 then
544:         Begin
545:           DataEvent.Wilayah[i].koordY:=DataGate.WilGate[NoGateUp].
546:             koordY+30+DataEvent.Wilayah[i].Tinggi div 2;
547:           IInc:=DataEvent.Count;
548:           Repeat
549:             Dec(IInc);
550:             Until DataEvent.Wilayah[IInc].NumberGateUp=NoGateUp;
551:           DataEvent.Wilayah[i].koordX:=DataEvent.Wilayah[IInc].koordX+
552:             DataEvent.Wilayah[IInc].Lebar div 2+10+DataEvent.Wilayah[i].
553:             Lebar div 2;
554:         end
555:       else
556:         Begin
557:           DataEvent.Wilayah[i].koordX:=DataGate.WilGate[NoGateUp].koordX;
558:           DataEvent.Wilayah[i].koordY:=DataGate.WilGate[NoGateUp].koordY+15
559:             +DataEvent.Wilayah[i].Tinggi div 2;
560:         end;

```

```

561: end
562: else
563:     Begin
564:         DataEvent.Wilayah[i].koordX:=320;
565:         DataEvent.Wilayah[i].koordY:=50;
566:     end;
567:     TengahinEvent;
568:     For IIInc:=1 to DataEvent.Count do
569:         CheckArea(IIInc, DataEvent.Wilayah[IIInc].KoordX, DataEvent.
570:             Wilayah[IIInc].KoordY);
571:         DarkI:=i;
572:         DarkGate:=0;
573:     end;
574:
575: procedure TForm2.ParameterGate(PosX, PosY, NoEventUp: integer;
576:     TextGate: String);
577: var
578:     ArcRectT: integer;
579:     ArcRectB: integer;
580:     StartArc: integer;
581:     AdGate: PRecGate;
582:     DRecGate: TRecGate;
583: Begin
584:     igate:=igate+1;
585:     if igate>DataGate.Count then
586:         Begin
587:             DRecGate:=TRecGate.create;
588:             DataGate.Expand;
589:             New(AdGate);
590:             AdGate^:=DRecGate;
591:             DataGate.Add(AdGate);
592:         end;
593:         ArcRectT:=PosY+DataEvent.Wilayah[NoEventUp].Tinggi div 2+15;
594:         ArcRectB:=ArcRectT+50;
595:         StartArc:=(ArcRectB-ArcRectT)div 2 + ArcRectT;
596:         DataGate.WilGate[igate].koordX:=PosX;
597:         DataGate.WilGate[igate].koordY:=StartArc;
598:         DataGate.WilGate[igate].Textnya:=TextGate;
599:         DataGate.WilGate[igate].NumberOfEvent:=0;
600:         DataGate.WilGate[igate].NumberEventUp:=NoEventUp;
601:         DataEvent.Wilayah[NoEventUp].NumberGateDown:=igate;
602:         DarkGate:=igate;
603:         DarkI:=0;
604:     end;
605:
606:
607: procedure TForm2.PaintBox1MouseDown(Sender: TObject; Button:
608:     TMouseButton; Shift: TShiftState; X, Y: Integer);
609: var
610:     IIInc: integer;
611:     IGateInc: integer;
612:     NoGateUp: integer;
613:     Gesernya: integer;
614:     OnEvent, OnGate: Boolean;
615:     Koordinat: TPoint;
616:     UpNumber : Integer;
617: Begin
618:     OnEvent:=False;
619:     OnGate:=False;
620:     if Button=mbLeft then
621:         Begin
622:             Drawing:=True;
623:             PaintBox1.Canvas.MoveTo(X, Y);
624:             IIInc:=0;
625:             UpNumber:=0;
626:             repeat
627:                 inc(IIInc);
628:                 If (X>=DataEvent.Wilayah[IIInc].koordX-50) and (X<=DataEvent.
629:                     Wilayah[IIInc].koordX+50) and (Y>=DataEvent.Wilayah[IIInc].koordY-
630:                     (DataEvent.Wilayah[IIInc].Tinggi div 2)) and (Y<=DataEvent.

```



```

631: Wilayah[IInc].koordY+(DataEvent.Wilayah[IInc].Tinggi div 2))
632: then begin
633:     OnEvent:=True;
634:     if ANDGatel.Checked=True then
635:         Begin
636:             if (DataEvent.Wilayah[IInc].NumberGateDown<>0) then
637:                 Begin
638:                     MessageDlg('There is a gate under this Event, you
639:                         can not add '+' more gate to this Event.',mtInformation,
640:                         [mbOK],0);
641:                 end
642:             else
643:                 Begin
644:                     X:=DataEvent.Wilayah[IInc].koordX;
645:                     Y:=DataEvent.Wilayah[IInc].koordY;
646:                     ParameterGate(X,Y,IInc,'AND');
647:                     ANDGatel.Checked:=False;
648:                     PaintBox1.Cursor:=crDefault;
649:                 end;
650:             end
651:         else if ORGatel.Checked=True then
652:             Begin
653:                 if (DataEvent.Wilayah[IInc].NumberGateDown<>0) then
654:                     Begin
655:                         MessageDlg('There is a gate under this Event, you can not
656:                             add '+' more gate to this Event.',mtInformation,[mbOK],0);
657:                     end
658:                 else
659:                     Begin
660:                         X:=DataEvent.Wilayah[IInc].koordX;
661:                         Y:=DataEvent.Wilayah[IInc].koordY;
662:                         ParameterGate(X,Y,IInc,'OR');
663:                         ORGatel.Checked:=False;
664:                         PaintBox1.Cursor:=crDefault;
665:                     end;
666:                 end
667:             else
668:                 Begin
669:                     Label2.Caption:=IntToStr(DataEvent.Wilayah[IInc].
670:                         NumberGateUp);
671:                     Label3.Caption:=IntToStr(DataEvent.Wilayah[IInc].
672:                         NumberGateDown);
673:                     Label4.Caption:=IntToStr(IInc);
674:                     PaintBox1.Cursor:=crIBeam;
675:                     DarkGate:=0;
676:                     DarkI:=IInc;
677:                 end;
678:                 UpNumber:=IInc;
679:             end;
680: Until IInc=DataEvent.Count;
681: IGateInc:=0;
682: repeat
683:     Inc(IGateInc);
684:     If (X>=DataGate.WilGate[IGateInc].koordX-25) and (X<=DataGate.
685:         WilGate[IGateInc].koordX+25) and (Y>=DataGate.WilGate[IGateInc].
686:         koordY-25)
687:     and (Y<=DataGate.WilGate[IGateInc].koordY) then
688:     begin
689:         OnGate:=True;
690:         if PrimeEvent1.Checked=True Then
691:             Begin
692:                 NoGateUp:=IGateInc;
693:                 PText:='Prime Event';
694:                 ParamEvent(NoGateUp,PText);
695:                 DataGate.WilGate[IGateInc].NumberOfEvent:=DataGate.
696:                     WilGate[IGateInc].NumberOfEvent+1;
697:                 PrimeEvent1.Checked:=False;
698:                 PaintBox1.Cursor:=crDefault;
699:             end
700:         else if IntermediateEvent1.Checked=True then

```



```

701:      Begin
702:          NoGateUp:=IGateInc;
703:          IText:='Inter Event';
704:          ParamEvent(NoGateUp, IText);
705:          DataGate.WilGate[IGateInc].NumberOfEvent:=DataGate.
706:          WilGate[IGateInc].NumberOfEvent+1;
707:          IntermediateEvent1.Checked:=False;
708:          PaintBox1.Cursor:=crDefault;
709:      end
710:  else
711:      Begin
712:          Label1.Caption:=IntToStr(DataGate.WilGate[IGateInc].
713:          NumberOfEvent);
714:          DarkI:=0;
715:          DarkGate:=IGateInc
716:      end;
717:  end;
718: Until IGateInc=DataGate.Count;
719: TengahinEvent;
720: {Mengatur lebar & tinggi notebook bila FTA melebihi lebar & tinggi
721: notebook, serta menggeser FTA bila FTA melebihi batas kiri notebook}
722: IInc:=0;
723: repeat
724:     inc(IInc);
725:     if ((DataEvent.Wilayah[IInc].KoordX-DataEvent.Wilayah[IInc].
726:     Lebar div 2-10)<0) then
727:         Begin
728:             Gesernya:=0-(DataEvent.Wilayah[IInc].KoordX-DataEvent.
729:             Wilayah[IInc].Lebar div 2-10);
730:             Redrawdown(1, Gesernya);
731:         end;
732:         if ((DataEvent.Wilayah[IInc].KoordX+DataEvent.Wilayah[IInc].
733:         Lebar div 2+10)>Notebook1.ClientWidth) then
734:             Begin
735:                 ScrollBox1.HorzScrollBar.Range:=ScrollBox1.HorzScrollBar.Range+
736:                 (DataEvent.Wilayah[IInc].KoordX+DataEvent.Wilayah[IInc].Lebar
737:                 div 2+10-Notebook1.ClientWidth);
738:                 Notebook1.ClientWidth:=ScrollBox1.HorzScrollBar.Range;
739:                 PaintBox1.ClientWidth:=NoteBook1.ClientWidth;
740:             end;
741:             if ((DataEvent.Wilayah[IInc].KoordY+DataEvent.Wilayah[IInc].Tinggi
742:             div 2+55)>Notebook1.ClientHeight) then
743:                 Begin
744:                     ScrollBox1.VertScrollBar.Range:=ScrollBox1.VertScrollBar.
745:                     Range+(DataEvent.Wilayah[IInc].KoordY+DataEvent.Wilayah[IInc].
746:                     Tinggi div 2+55-Notebook1.ClientHeight);
747:                     Notebook1.ClientHeight:=ScrollBox1.VertScrollBar.Range;
748:                     PaintBox1.ClientHeight:=NoteBook1.ClientHeight;
749:                 end;
750: Until IInc=DataEvent.Count;
751: ScrollToDark;
752: end
753: else if Button=mbRight then
754: Begin
755:     PaintBox1.Canvas.MoveTo(X, Y);
756:     For IInc:=1 to DataEvent.Count do
757:         Begin
758:             If (X>=DataEvent.Wilayah[IInc].koordX-50) and (X<=DataEvent.
759:             Wilayah[IInc].koordX+50) and (Y>=DataEvent.Wilayah[IInc].koordY-
760:             (DataEvent.Wilayah[IInc].Tinggi div 2)) and (Y<=DataEvent.
761:             Wilayah[IInc].koordY+(DataEvent.Wilayah[IInc].Tinggi div 2))
762:             then
763:                 Begin
764:                     OnEvent:=True;
765:                     DarkI:=IInc;
766:                     DarkGate:=0;
767:                     PaintBox1.Refresh;
768:                     ScrollToDark;
769:                     Koordinat:=PaintBox1.ClientToScreen(Point(X, Y));
770:                     PopUpMenu1.PopUp(Koordinat.X, Koordinat.Y);

```

```

771:     end;
772: end;
773: For IGateInc:=1 to DataGate.Count do
774: Begin
775: If (X>=DataGate.WilGate[IGateInc].koordX-25) and (X<=DataGate.
776:   WilGate[IGateInc].koordX+25) and (Y>=DataGate.WilGate[IGateInc].
777:   koordY-25) and (Y<=DataGate.WilGate[IGateInc].koordY)
778: then
779:   Begin
780:     OnGate:=True;
781:     DarkGate:=IGateInc;
782:     DarkI:=0;
783:     PaintBox1.Refresh;
784:     ScrollToDark;
785:     Koordinat:=PaintBox1.ClientToScreen(Point(X,Y));
786:     PopUpMenu1.Popup(Koordinat.X,Koordinat.Y);
787:   end;
788: end;
789: end;
790: If (OnEvent=False) and (OnGate=False) then
791: Begin
792:   DarkI:=0;
793:   DarkGate:=0;
794: end;
795: For IIInc:=1 to DataEvent.Count do
796:   CheckArea(IIInc,DataEvent.Wilayah[IIInc].KoordX,DataEvent.
797:   Wilayah[IIInc].KoordY);
798: TengahinEvent;
799: PaintBox1.Refresh;
800: end;
801:
802:
803: procedure TForm2.ScrollToDark;
804: Begin
805: {Mengatur posisi scrollbar bila Gate atau Event diblok}
806: if (DarkGate<>0) then
807:   Begin
808:     if (DataGate.WilGate[DarkGate].koordx>640) then
809:       ScrollBox1.HorzScrollBar.Position:=DataGate.WilGate[DarkGate].
810:       koordx-320;
811:     if (DataGate.WilGate[DarkGate].koordY>320) then
812:       ScrollBox1.VertScrollBar.Position:=DataGate.WilGate[DarkGate].
813:       koordY-160;
814:   end;
815: if (DarkI<>0) then
816:   Begin
817:     if (DataEvent.Wilayah[DarkI].koordx+DataEvent.Wilayah[DarkI].
818:     lebar div 2 +5>640)
819:     then
820:       ScrollBox1.HorzScrollBar.Position:=DataEvent.Wilayah[DarkI].
821:       koordx-320;
822:     if (DataEvent.Wilayah[DarkI].koordY>320) then
823:       ScrollBox1.VertScrollBar.Position:=DataEvent.Wilayah[DarkI].
824:       koordY-160;
825:   end;
826: end;
827:
828:
829: procedure TForm2.FormDestroy(Sender: TObject);
830: begin
831: DataEvent.Clear;
832: end;
833:
834: procedure TForm2.FormActivate(Sender: TObject);
835: begin
836:   TEvent:=True;
837:   IEvent:=True;
838:   PEvent:=True;
839:   AButton:=True;
840:   OButton:=True;

```



```

841: end;
842:
843: procedure TForm2.DelGate(DarkGate: integer);
844: var
845:     IInc, IGateInc: integer;
846: Begin
847: DataEvent.Wilayah[DataGate.WilGate[DarkGate].NumberEventUp].
848: NumberGateDown:=0;
849: IGateInc:=DarkGate;
850: While IGateInc<DataGate.Count do
851: Begin
852:     inc(IGateInc);
853:     For IInc:=1 to DataEvent.Count do
854:         Begin
855:             if DataEvent.Wilayah[IInc].NumberGateDown=IGateInc then
856:                 DataEvent.Wilayah[IInc].NumberGateDown:=IGateInc-1
857:             else if DataEvent.Wilayah[IInc].NumberGateUp=IGateInc then
858:                 DataEvent.Wilayah[IInc].NumberGateUp:=IGateInc-1;
859:         end;
860:     end;
861: DataGate.Delete(DarkGate-1);
862: igate:=igate-1;
863: end;
864:
865:
866: procedure TForm2.DelEvent(DarkI: integer);
867: var
868:     IInc, IGateInc: integer;
869:     NoEvent, NoGate: integer;
870: Begin
871: NoEvent:=DarkI;
872: NoGate:=DataEvent.Wilayah[Noevent].NumberGateUp;
873: while NoGate>=1 do
874: Begin
875:     For IInc:=1 to DataEvent.Count do
876:         Begin
877:             If DataEvent.Wilayah[IInc].NumberGateUp=DataEvent.
878:                 Wilayah[NoEvent].NumberGateUp then
879:                 Begin
880:                     If IInc<NoEvent then
881:                         Redrawdown(IInc,DataEvent.Wilayah[NoEvent].Lebar div 2+5)
882:                     else if IInc>NoEvent then
883:                         Redrawdown(IInc,-(DataEvent.Wilayah[NoEvent].Lebar div 2+5));
884:                 end;
885:             end;
886:             NoEvent:=DataGate.WilGate[NoGate].NumberEventUp;
887:             NoGate:=DataEvent.Wilayah[NoEvent].NumberGateUp;
888:         end;
889: DataGate.WilGate[DataEvent.Wilayah[DarkI].NumberGateUp].
890: NumberOfEvent:=DataGate.WilGate[DataEvent.Wilayah[DarkI].
891:     NumberGateUp].NumberOfEvent-1;
892: IInc:=DarkI;
893: While IInc<DataEvent.Count do
894: Begin
895:     inc(IInc);
896:     For IGateInc:=1 to DataGate.Count do
897:         Begin
898:             if DataGate.WilGate[IGateInc].NumberEventUp=IInc then
899:                 DataGate.WilGate[IGateInc].NumberEventUp:=IInc-1
900:         end;
901:     end;
902: DataEvent.Delete(DarkI-1);
903: i:=i-1;
904: TengahinEvent;
905: For IInc:=1 to DataEvent.Count do
906: CheckArea(IInc,DataEvent.Wilayah[IInc].KoordX,DataEvent.
907: Wilayah[IInc].KoordY);
908: end;
909:
910:

```



```

911: procedure TForm2.DelDownGate(GateTop: integer);
912: var
913:   IInc: integer;
914: Begin
915:   if DataGate.WilGate[GateTop].NumberOfEvent=0 then Delgate(GateTop)
916:   else
917:   Begin
918:     IInc:=DataEvent.Count;
919:     Repeat
920:       If DataEvent.Wilayah[IInc].NumberGateUp=GateTop then
921:       Begin
922:         If DataEvent.Wilayah[IInc].NumberGateDown=0
923:         then DelEvent(IInc)
924:         else
925:         Begin
926:           DelDownGate(DataEvent.Wilayah[IInc].NumberGateDown);
927:           DelEvent(IInc);
928:         end;
929:       end;
930:       Dec(IInc);
931:     Until IInc=0;
932:     Delgate(GateTop);
933:   end;
934: end;
935:
936:
937: procedure TForm2.DelDownEvent(EventTop: integer);
938: Begin
939:   If DataEvent.Wilayah[EventTop].NumberGateDown=0 then
940:     DelEvent(EventTop)
941:   else
942:   Begin
943:     DelDownGate(DataEvent.Wilayah[EventTop].NumberGateDown);
944:     DelEvent(EventTop);
945:   end;
946: end;
947:
948:
949: procedure TForm2.TengahinEvent;
950: var
951:   IInc,IGateInc: integer;
952:   XFirst,XEnd,XCentre,GeserCentre: integer;
953: Begin
954:   {Event-event di-centre-kan terhadap gate diatasnya}
955:   if DataGate.Count<>0 then
956:   Begin
957:     IGateInc:=0;
958:     repeat
959:       inc(IGateInc);
960:       IInc:=0;
961:       XFirst:=0;
962:       XEnd:=0;
963:       repeat
964:         inc(IInc);
965:         if(DataEvent.Wilayah[IInc].NumberGateUp=IGateInc) then
966:         Begin
967:           if(DataEvent.Wilayah[IInc].KoordX<XFirst)or(XFirst=0) then
968:             XFirst:=DataEvent.Wilayah[IInc].KoordX;
969:           if(DataEvent.Wilayah[IInc].KoordX>XEnd) then
970:             XEnd:=DataEvent.Wilayah[IInc].KoordX;
971:         end;
972:       Until IInc=DataEvent.Count;
973:       XCentre:=(XEnd-XFirst)div 2+XFirst;
974:       GeserCentre:=DataGate.WilGate[IGateInc].KoordX-XCentre;
975:       IInc:=0;
976:       repeat
977:         inc(IInc);
978:         if(DataEvent.Wilayah[IInc].NumberGateUp=IGateInc) then
979:           RedrawDown(IInc,GeserCentre);
980:       Until IInc=DataEvent.Count;

```

```
981:    Until IGateInc=DataGate.Count;
982:    end;
983: end;
984:
985: procedure TForm2.SetCursor;
986: begin
987:   If (TEvent=True) or (PEvent=True) or (IEvent=True) or (AButton=True) or
988:     (OButton=True) Then
989:     PaintBox1.Cursor:=crCross
990:   else
991:     PaintBox1.Cursor:=crDefault;
992: end;
993:
994: procedure TForm2.TopEvent1Click(Sender: TObject);
995: begin
996:   ORGate1.Checked:=False;
997:   ANDGate1.Checked:=False;
998:   PrimeEvent1.Checked:=False;
999:   IntermediateEvent1.Checked:=False;
1000:   TopEvent1.Checked:=True;
1001:   TEvent:=True;
1002:   IEvent:=False;
1003:   PEvent:=False;
1004:   OButton:=False;
1005:   AButton:=False;
1006:   SetCursor;
1007: end;
1008:
1009: procedure TForm2.IntermediateEvent1Click(Sender: TObject);
1010: begin
1011:   ORGate1.Checked:=False;
1012:   ANDGate1.Checked:=False;
1013:   PrimeEvent1.Checked:=False;
1014:   IntermediateEvent1.Checked:=True;
1015:   TopEvent1.Checked:=False;
1016:   TEvent:=False;
1017:   IEvent:=True;
1018:   PEvent:=False;
1019:   OButton:=False;
1020:   AButton:=False;
1021:   SetCursor;
1022: end;
1023:
1024: procedure TForm2.Primeevent1Click(Sender: TObject);
1025: begin
1026:   ORGate1.Checked:=False;
1027:   ANDGate1.Checked:=False;
1028:   PrimeEvent1.Checked:=True;
1029:   IntermediateEvent1.Checked:=False;
1030:   TopEvent1.Checked:=False;
1031:   TEvent:=False;;
1032:   IEvent:=False;
1033:   PEvent:=True;
1034:   OButton:=False;
1035:   AButton:=False;
1036:   SetCursor;
1037: end;
1038:
1039: procedure TForm2.ORGate1Click(Sender: TObject);
1040: begin
1041:   ORGate1.Checked:=True;
1042:   ANDGate1.Checked:=False;
1043:   PrimeEvent1.Checked:=False;
1044:   IntermediateEvent1.Checked:=False;
1045:   TopEvent1.Checked:=False;
1046:   TEvent:=False;
1047:   IEvent:=False;
1048:   PEvent:=False;
1049:   OButton:=True;
1050:   AButton:=False;
```



```

1051:   SetCursor;
1052: end;
1053:
1054: procedure TForm2.ANDGate1Click(Sender: TObject);
1055: begin
1056:   ORGate1.Checked:=False;
1057:   ANDGate1.Checked:=True;
1058:   PrimeEvent1.Checked:=False;
1059:   IntermediateEvent1.Checked:=False;
1060:   TopEvent1.Checked:=False;
1061:   TEvent:=False;;
1062:   IEvent:=False;
1063:   PEvent:=False;
1064:   OButton:=False;
1065:   AButton:=True;
1066:   SetCursor;
1067: end;
1068:
1069: procedure TForm2.Delete2Click(Sender: TObject);
1070: begin
1071:   If DarkI<>0 then
1072:     DelDownEvent(DarkI)
1073:   else if DarkGate<>0 then
1074:     DelDownGate(DarkGate);
1075:   TengahinEvent;
1076:   PaintBox1.Refresh;
1077: end;
1078:
1079: procedure TForm2.PaintBox1Paint(Sender: TObject);
1080: var
1081:   Tulisan: TRect;
1082:   IInc, IGateInc: integer;
1083:   KoordL, KoordR, KoordT, KoordB: integer;
1084:   ArcRectL, ArcRectR, ArcRectT, ArcRectB, StartArc: integer;
1085: begin
1086:   PaintBox1.Canvas.Pen.Color:=clBlack;
1087:   For IInc:=1 to DataEvent.Count do
1088:   Begin
1089:     if IInc=DarkI then
1090:     Begin
1091:       PaintBox1.Canvas.Brush.Color:=clBlack;
1092:       PaintBox1.Canvas.Font.Color:=clWhite;
1093:     end
1094:     else
1095:     Begin
1096:       PaintBox1.Canvas.Brush.Color:=clWhite;
1097:       PaintBox1.Canvas.Font.Color:=clBlack;
1098:     end;
1099:     KoordL:=DataEvent.Wilayah[IInc].koordx-DataEvent.Wilayah[IInc].
1100:       Lebar div 2;
1101:     KoordR:=DataEvent.Wilayah[IInc].koordx+DataEvent.Wilayah[IInc].
1102:       Lebar div 2;
1103:     KoordT:=DataEvent.Wilayah[IInc].koordY-DataEvent.Wilayah[IInc].
1104:       Tinggi div 2;
1105:     KoordB:=DataEvent.Wilayah[IInc].koordY+DataEvent.Wilayah[IInc].
1106:       Tinggi div 2;
1107:     Tulisan.Top := KoordT;
1108:     Tulisan.Left := KoordL;
1109:     Tulisan.Bottom := KoordB;
1110:     Tulisan.Right := KoordR;
1111:     PaintBox1.Canvas.Rectangle(KoordL, KoordT, KoordR, KoordB);
1112:     PaintBox1.Canvas.TextOut(KoordL+25, KoordT+15, DataEvent.
1113:       Wilayah[IInc].Textnya);
1114:     If (DataEvent.Wilayah[IInc].TypeEvent='Inter Event') or(DataEvent.
1115:       Wilayah[IInc].TypeEvent='Prime Event') then
1116:       PaintBox1.Canvas.Polyline([Point(DataEvent.Wilayah[IInc].koordx,
1117:         KoordT), Point(DataEvent.Wilayah[IInc].koordx, DataEvent.
1118:         Wilayah[IInc].koordY-DataEvent.Wilayah[IInc].Tinggi div 2-15)]);
1119:   end;
1120:   For IGateInc:=1 to DataGate.Count do

```




```

1121: Begin
1122:   ArcRectL:=DataGate.WilGate[IGateInc].koordx-25;
1123:   ArcRectT:=DataGate.WilGate[IGateInc].koordY-25;
1124:   ArcRectR:=DataGate.WilGate[IGateInc].koordX+25;
1125:   ArcRectB:=DataGate.WilGate[IGateInc].koordY+25;
1126:   StartArc:=(ArcRectB-ArcRectT)div 2 + ArcRectT;
1127:   PaintBox1.Canvas.Arc(ArcRectL,ArcRectT,ArcRectR,ArcRectB,ArcRectR,
1128:   StartArc,ArcRectL,StartArc);
1129:   PaintBox1.Canvas.Polyline([Point(ArcRectL,StartArc)
1130:   ,Point(ArcRectR,StartArc)]);
1131:   if IGateInc=DarkGate then
1132:     Begin
1133:       PaintBox1.Canvas.Brush.Color:=clBlack;
1134:       PaintBox1.Canvas.Font.Color:=clWhite;
1135:       PaintBox1.Canvas.FloodFill(ArcRectL+1, StartArc-1, clBlack,
1136:       fsBorder);
1137:     end
1138:   else
1139:     Begin
1140:       PaintBox1.Canvas.Brush.Color:=clWhite;
1141:       PaintBox1.Canvas.Font.Color:=clBlack;
1142:       PaintBox1.Canvas.FloodFill(ArcRectL+1, StartArc-1, clWhite,
1143:       fsBorder);
1144:     end;
1145:     if (DataGate.WilGate[IGateInc].Textnya='AND') then
1146:       PaintBox1.Canvas.TextOut(ArcRectL+11,StartArc-18,DataGate.
1147:       WilGate[IGateInc].Textnya)
1148:     else
1149:       PaintBox1.Canvas.TextOut(ArcRectL+15,StartArc-18,DataGate.
1150:       WilGate[IGateInc].Textnya);
1151:       PaintBox1.Canvas.Polyline([Point(DataGate.WilGate[IGateInc].
1152:       koordX,DataEvent.Wilayah[DataGate.WilGate[IGateInc].
1153:       NumberEventUp].koordY+DataEvent.Wilayah[DataGate.
1154:       WilGate[IGateInc].NumberEventUp].Tinggi div 2),Point(DataGate.
1155:       WilGate[IGateInc].koordX,ArcRectT)]);
1156:       PaintBox1.Canvas.Polyline([Point(DataGate.WilGate[IGateInc].
1157:       koordx,DataGate.WilGate[IGateInc].koordY),Point(DataGate.
1158:       WilGate[IGateInc].koordx,DataGate.WilGate[IGateInc].koordY+15)]);
1159:       if (DataGate.WilGate[IGateInc].NumberOfEvent<>0) then
1160:         Begin
1161:           For IInc:=1 to DataEvent.Count do
1162:             Begin
1163:               if DataEvent.Wilayah[IInc].NumberGateUp=IGateInc then
1164:                 PaintBox1.Canvas.Polyline([Point(DataGate.WilGate[IGateInc].
1165:                 koordx,DataGate.WilGate[IGateInc].koordY+15),Point(DataEvent.
1166:                 Wilayah[IInc].koordx,DataEvent.Wilayah[IInc].koordY-DataEvent.
1167:                 Wilayah[IInc].Tinggi div 2-15)]);
1168:             end;
1169:           end;
1170:         end;
1171:       end;
1172:     end;
1173:   end;
1174: procedure TForm2.Properties2Click(Sender: TObject);
1175: var
1176:   CB1,CB2,CB3, RBG : string;
1177:   {Koordinat: TPoint;}
1178: begin
1179:   if (DarkI<>0) then
1180:     Begin
1181:       CB1:=DataEvent.Wilayah[DarkI].Textnya;
1182:       CB2:=DataEvent.Wilayah[DarkI].TypeEvent;
1183:       CB3:=FloatToStr(DataEvent.Wilayah[DarkI].Probability);
1184:       if Form3.GetString(CB1,CB2,CB3) then
1185:         Begin
1186:           DataEvent.Wilayah[DarkI].Textnya:=CB1;
1187:           DataEvent.Wilayah[DarkI].TypeEvent:=CB2;
1188:           DataEvent.Wilayah[DarkI].Probability:=StrToFloat(CB3);
1189:           ReCalculate;
1190:         end;

```

```

1191: end
1192: else if (DarkGate<>0) then
1193: Begin
1194:   RBG:=DataGate.WilGate[DarkGate].Textnya;
1195:   if Form4.GetString(RBG) then
1196:     Begin
1197:       DataGate.WilGate[DarkGate].Textnya:=RBG;
1198:       ReCalculate;
1199:     end;
1200: end;
1201: end;
1202:
1203: procedure TForm2.Properties1Click(Sender: TObject);
1204: var
1205:   CB1,CB2,CB3, RBG: string;
1206:   {Koordinat: TPoint;}
1207: begin
1208:   if (DarkI<>0) then
1209:     Begin
1210:       CB1:=DataEvent.Wilayah[DarkI].Textnya;
1211:       CB2:=DataEvent.Wilayah[DarkI].TypeEvent;
1212:       CB3:=FloatToStr(DataEvent.Wilayah[DarkI].Probability);
1213:       if Form3.GetString(CB1,CB2,CB3) then
1214:         Begin
1215:           DataEvent.Wilayah[DarkI].Textnya:=CB1;
1216:           DataEvent.Wilayah[DarkI].TypeEvent:=CB2;
1217:           DataEvent.Wilayah[DarkI].Probability:=StrToFloat(CB3);
1218:           ReCalculate;
1219:         end;
1220:       end
1221:     else if (DarkGate<>0) then
1222:       Begin
1223:         RBG:=DataGate.WilGate[DarkGate].Textnya;
1224:         if Form4.GetString(RBG) then
1225:           DataGate.WilGate[DarkGate].Textnya:=RBG;
1226:           ReCalculate;
1227:         end;
1228:       end;
1229:
1230: procedure TForm2.Cut1Click(Sender: TObject);
1231: begin
1232:   if DarkI<>0 then
1233:     DelDownEvent(DarkI)
1234:   else if DarkGate<>0 then
1235:     DelDownGate(DarkGate);
1236:   TengahinEvent;
1237:   PaintBox1.Refresh;
1238: end;
1239:
1240: procedure TForm2.Cut2Click(Sender: TObject);
1241: begin
1242:   if DarkI<>0 then
1243:     DelDownEvent(DarkI)
1244:   else if DarkGate<>0 then
1245:     DelDownGate(DarkGate);
1246:   TengahinEvent;
1247:   PaintBox1.Refresh;
1248: end;
1249:
1250: procedure TForm2.PaintBox1DblClick(Sender: TObject);
1251: var
1252:   CB1,CB2,CB3, RBG: string;
1253:   {Koordinat: TPoint;}
1254: begin
1255:   if (DarkI<>0) then
1256:     Begin
1257:       CB1:=DataEvent.Wilayah[DarkI].Textnya;
1258:       CB2:=DataEvent.Wilayah[DarkI].TypeEvent;
1259:       CB3:=FloatToStr(DataEvent.Wilayah[DarkI].Probability);
1260:       if Form3.GetString(CB1,CB2,CB3) then

```



```

1: unit Unit6;
2:
3: interface
4:
5: uses
6:   SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
7:   Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
8:
9: type
10:   TForm3 = class(TForm)
11:     Label1: TLabel;
12:     Label3: TLabel;
13:     OK: TButton;
14:     Cancel: TButton;
15:     RadioGroup1: TRadioGroup;
16:     RadioButton1: TRadioButton;
17:     RadioButton2: TRadioButton;
18:     RadioButton3: TRadioButton;
19:     Edit1: TEdit;
20:     Edit2: TEdit;
21:     procedure RadioButton3Click(Sender: TObject);
22:     procedure RadioButton2Click(Sender: TObject);
23:   private
24:     function Execute: Boolean;
25:     { Private declarations }
26:   public
27:     function GetString(var CB1,CB2, CB3: string): Boolean;
28:     { Public declarations }
29:   end;
30:
31: var
32:   Form3: TForm3;
33:
34: implementation
35:
36: {$R *.DFM}
37:
38: function TForm3.Execute: Boolean;
39: begin
40:   Result := ShowModal = mrOk;
41: end;
42:
43: function TForm3.GetString(var CB1,CB2, CB3 : string): Boolean;
44: begin
45:   Edit1.Text := CB1;
46:   Edit2.Text := CB3;
47:   if CB2='Inter Event' then Begin
48:     RadioButton2.Enabled:=True;
49:     RadioButton3.Enabled:=True;
50:     RadioButton1.Enabled:=False;
51:     RadioButton2.Checked:=True;
52:     Edit2.Enabled:=False;
53:   End
54:   else if CB2='Prime Event' then Begin
55:     RadioButton2.Enabled:=True;
56:     RadioButton3.Enabled:=True;
57:     RadioButton1.Enabled:=False;
58:     RadioButton3.Checked:=True;
59:     Edit2.Enabled:=True;
60:   End
61:   else if CB2='Top Event' then Begin
62:     RadioButton1.Enabled:=True;
63:     RadioButton2.Enabled:=False;
64:     RadioButton3.Enabled:=False;
65:     RadioButton1.Checked:=True;
66:     Edit2.Enabled:=False;
67:   End;
68:   Result := Form3.Execute;
69:   if Result then
70:     Begin

```



```
71:      CB1:=Edit1.Text;
72:      If RadioButton2.Checked=True Then CB2:='Inter Event'
73:      Else If RadioButton3.Checked=True Then CB2:='Prime Event';
74:      CB3:=Edit2.Text;
75:  end;
76: end;
77:
78: procedure TForm3.RadioButton3Click(Sender: TObject);
79: begin
80:   Label3.Enabled:=True;
81: end;
82:
83: procedure TForm3.RadioButton2Click(Sender: TObject);
84: begin
85:   Label3.Enabled:=False;
86: end;
87:
88:
89: end.
```

```
1: unit Unit7;
2:
3: interface
4:
5: uses
6:   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
7:   Dialogs, StdCtrls, ExtCtrls;
8:
9: type
10:   TForm4 = class(TForm)
11:     OK: TButton;
12:     Cancel: TButton;
13:     RadioButton1: TRadioButton;
14:     RadioButton2: TRadioButton;
15:   private
16:     function Execute: Boolean;
17:     { Private declarations }
18:   public
19:     function GetString(var RBG: string): Boolean;
20:     { Public declarations }
21:   end;
22:
23: var
24:   Form4: TForm4;
25:
26: implementation
27:
28: {$R *.DFM}
29:
30: function TForm4.Execute: Boolean;
31: begin
32:   Result := ShowModal = mrOk;
33: end;
34:
35: function TForm4.GetString(var RBG: string): Boolean;
36: begin
37:   if RBG='AND' then
38:     RadioButton1.Checked := True
39:   else if RBG='OR' then
40:     RadioButton2.Checked := True;
41:   Result := Form4.Execute;
42:   if Result then
43:     begin
44:       if RadioButton1.Checked=True then
45:         RBG := 'AND'
46:       else
47:         RBG := 'OR';
48:     end;
49:   end;
50:
51: end.
```

```

1: unit Unit8;
2:
3: interface
4:
5: uses
6:   SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
7:   Controls, Forms, Dialogs, Menus, Buttons, ExtCtrls, ComCtrls,
8:   ToolWin, Unit2;
9:
10: type
11:   TForm1 = class(TForm)
12:     MainMenu1: TMainMenu;
13:     File1: TMenuItem;
14:     Help1: TMenuItem;
15:     AboutRel1: TMenuItem;
16:     New1: TMenuItem;
17:     Open1: TMenuItem;
18:     N1: TMenuItem;
19:     Exit1: TMenuItem;
20:     OpenDialog1: TOpenDialog;
21:     ToolBar1: TToolBar;
22:     NewToolButton: TToolButton;
23:     ImageList1: TImageList;
24:     OpenToolButton: TToolButton;
25:     ToolButton3: TToolButton;
26:     TEToolButton: TToolButton;
27:     IEToolButton: TToolButton;
28:     PEToolButton: TToolButton;
29:     ANDToolButton: TToolButton;
30:     ORToolButton: TToolButton;
31:     procedure Exit1Click(Sender: TObject);
32:     procedure Open1Click(Sender: TObject);
33:     procedure OpenToolButtonClick(Sender: TObject);
34:     procedure TEToolButtonClick(Sender: TObject);
35:     procedure PEToolButtonClick(Sender: TObject);
36:     procedure IEToolButtonClick(Sender: TObject);
37:     procedure ANDToolButtonClick(Sender: TObject);
38:     procedure ORToolButtonClick(Sender: TObject);
39:     procedure FormShow(Sender: TObject);
40:     procedure NewToolButtonClick(Sender: TObject);
41:     procedure FormShow(Sender: TObject);
42:     procedure NewToolButtonClick(Sender: TObject);
43:   private
44:     { Private declarations }
45:   public
46:     { Public declarations }
47:   end;
48:
49: var
50:   Form1: TForm1;
51:
52: implementation
53:
54: {$R *.DFM}
55:
56: var
57:   Form2: TForm2;
58:
59: procedure TForm1.Exit1Click(Sender: TObject);
60: begin
61:   Close;
62: end;
63:
64: procedure TForm1.Open1Click(Sender: TObject);
65: begin
66:   OpenDialog1.Execute

```



```
67: end;
68:
69: procedure TForm1.OpenToolButtonClick(Sender: TObject);
70: begin
71:   OpenDialog1.Execute
72: end;
73:
74: procedure TForm1.TEToolButtonClick(Sender: TObject);
75: begin
76:   Form2.SetCursor;
77: end;
78:
79: procedure TForm1.PEToolButtonClick(Sender: TObject);
80: begin
81:   Form2.SetCursor;
82: end;
83:
84: procedure TForm1.IEToolButtonClick(Sender: TObject);
85: begin
86:   Form2.SetCursor;
87: end;
88:
89: procedure TForm1.ANDToolButtonClick(Sender: TObject);
90: begin
91:   Form2.SetCursor;
92: end;
93:
94: procedure TForm1.ORToolButtonClick(Sender: TObject);
95: begin
96:   Form2.SetCursor;
97: end;
98:
99: procedure TForm1.New1Click(Sender: TObject);
100: begin
101:   BikinMDIChild('Untitled-' + IntToStr(MDIChildCount + 1));
102: end;
103:
104: procedure TForm1.BikinMDIChild(const Name: string);
105: begin
106:   Form2:=TForm2.create(Application);
107:   Form2.Caption := Name;
108: end;
109:
110: procedure TForm1.FormShow(Sender: TObject);
111: begin
112:   if MDIChildCount=0 then
113:   Begin
114:     TEToolButton.Enabled:=False;
115:     IEToolButton.Enabled:=False;
116:     PEToolButton.Enabled:=False;
117:     ANDToolButton.Enabled:=False;
118:     ORToolButton.Enabled:=False;
119:   end;
120: end;
121:
122: procedure TForm1.NewToolButtonClick(Sender: TObject);
123: begin
124:   New1Click(Sender);
125: end;
126:
127: end.
```